

INTERVAL RESTRICTION.

A composition system and its automation.

by

Jirrah Walker MacArthur (B.A. Hons.)

Submitted in partial fulfilment of the
Requirements for the degree of
Master of Music (Research)

Conservatorium of Music

University of Tasmania

December 1999

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgment is made in the text of the thesis.

This thesis may be made available for loan and limited copying in accordance with the *Copyright Act 1968*.

A handwritten signature in black ink, reading 'J MacArthur'. The 'J' is large and loops around the 'M'. The 'A' is also large and loops around the 'c'. The signature is written in a cursive, flowing style.

Jirrah MacArthur

ABSTRACT

This dissertation presents composition project that includes a software tool and a folio of scores and recordings. A composition method called Interval Restriction Composition (IRC) is presented. The author has devised IRC for writing music using the equal tempered chromatic, and the system comes under close examination for its musical utility. A detailed study reveals that while IRC would allow the composer to write music with a high degree of intervallic integrity, the sheer volume of repetitive work needed to conform to the method places it out of reach of a composer working without the aid of a computer. The field of Computer Assisted Composition is then surveyed to determine whether available composition tools are suitable for the automation of IRC. It is acknowledged that there are tools that perform similar functions to that demanded of IRC, however they are considered to be either inadequate or inappropriate. Consequently, a software tool called Restricted Interval Counterpoint Engine (RICE) is presented. RICE is custom designed to automate Interval Restriction Composition. RICE also extends the system by providing musically useful functions, including auditioning. A folio of RICE assisted compositions and recordings is presented. These compositions are discussed in the context of RICE and IRC, as well as on their own merits. This discussion includes an evaluation of the compositions and the impact of RICE on IRC and the compositional process. Finally, discussion of future directions for the technical and musical improvement of IRC and the software conclude the dissertation.

ACKNOWLEDGMENTS

Eternal gratitude is reciprocated to Stephanie Ryan.

Thanks to my supervisor Raffæle Marcellino.

Thanks to Kim Bastin for sensitive and timely arrangement of *Four Fugues*.

Table of Contents

<u>1. INTRODUCTION</u>	<u>2</u>
<u>2. IRC. INTERVAL RESTRICTION COMPOSITION</u>	<u>5</u>
FIGURE 2.1 INTERVALLIC ANALYSIS.	7
FIGURE 2.2 INTERVAL RESTRICTION TABLES	8
FIGURE 2.3 SEARCHING FOR POSSIBLE SETS.	9
FIGURE 2.6 INTERVAL PERMUTATIONS APPLIED TO DIRECTION VECTORS.	16
FIGURE 2.7 SCHEMATIC REPRESENTATION OF IRC	17
FIGURE 2.8 HARMONIC INTERVALS AGAINST NUMBER OF VOICES.	18
<u>3. A SURVEY OF COMPUTER ASSISTED COMPOSITION TOOLS</u>	<u>21</u>
<u>4. RICE. RESTRICTED INTERVAL COUNTERPOINT ENGINE</u>	<u>32</u>
FIGURE 4.1 RICE'S INTERVAL RESTRICTIONS TABLES.	33
FIGURE 4.2 RICE'S PITCH CLASS RESTRICTIONS TABLE.	34
FIGURE 4.3 AN EXAMPLE OF AN EXPONENTIALLY RESIZED HARMONIC-INTERVALS TABLE.	35
FIGURE 4.4 RICE'S PIANO ROLL DISPLAY FOR MUSICAL DATA.	36
FIGURE 4.5 FLOW CHART FOR RICE'S SEARCH ENGINE.	38
FIGURE 4.5 EXTRACT A.	39
FIGURE 4.5 EXTRACT B.	39
FIGURE 4.5 EXTRACT C.	40
FIGURE 4.5 EXTRACT D.	41
FIGURE 4.5 EXTRACT E.	41
FIGURE 4.6 A PALETTE OF POSSIBLE SETS.	42

FIGURE 4.7 A DIRECTIONAL SEARCH.	43
FIGURE 4.8 THE RESTRICTION SET MANAGER WINDOW.	46
FIGURE 4.9 THE CANON MANAGER.	49
FIGURE 4.10 AN EXAMPLE OF RICE'S TEXT FILE OUTPUT.	51
 5. MUSICAL APPLICATION OF RICE	 53
 5.1 THREE SONGS FOR SOPRANO AND PIANO	 53
FIGURE 5.1 FORMAL DIVISIONS IN MORNING SONG.	59
FIGURE 5.2 FORMAL DIVISIONS IN THE NIGHT DANCES.	63
5.2 TEN PRAYERS FOR MIXED CHOIR.	67
FIGURE 5.3 VOICING OF TEN PRAYERS FOR MIXED CHOIR.	67
FIGURE 5.4. A RESTRICTION SET USED IN GOD BLESS THIS TINY LITTLE BOAT.	68
5.3 FOUR FUGUES	72
5.4 THREE REHARMONIZED JAZZ STANDARDS	77
 6 DISCUSSION AND SUMMARY	 79
 REFERENCES	 89
 APPENDIX. RICE SOFTWARE AND MANUAL	 99

1. INTRODUCTION

This dissertation presents a composition project. It details a composition method called Interval Restriction Composition. Computer automation of this method is described, and a supporting folio of music from the application of the method is presented. The musical worth of the composition method and its automation are then evaluated. This introductory chapter provides an overview of the dissertation's content.

Chapter 2 presents Interval Restriction Composition (IRC). IRC is a conceptually simple method of maintaining intervallic integrity on the surface-level of a composition. To limit and unify the sound of the composition or parts thereof, the composer draws up tables of intervals that are permitted linearly and vertically. These tables apply to each voice, and in between each voice-pair in the composition. The composer then plots a musical 'path' through these restrictions ensuring that each new musical event conforms to the interval tables. This method achieves surface-level intervallic unity, and allows for flexible evolution of the intervallic sound during composition through the alteration of the restriction tables.

This simple method is impractical. The number of possible musical paths that the composer must test against the interval tables for a single musical event is huge. This number grows exponentially in proportion to the number of voices in the composition and the number of intervals in the tables. For a composition of reasonable complexity, a short chord progression would take an impractical length of time to calculate.

The problem with IRC is that the workload required of the composer is unreasonably high. Massive searches must be solved accurately, and this is an extremely time consuming and dull task. Searching interval space and storing the solutions is a repetitive and routine task ideally suited to computer automation.

The computer automation of musical tasks falls within the field of Computer Assisted Composition, and Chapter 3 surveys this field. The field of Computer Assisted Composition (CAC) is very diverse, including all possible ways in which

a computer may be employed during the compositional process, from notation, audio synthesis through to automatic music generation. For the purposes of this dissertation, CAC software tools are defined as tools which help the composer generate precompositional material in the equal tempered chromatic. Chapter 3 reviews software tools which fall within this scope, and assesses them for their suitability for application to the automation of IRC. Although there are a number of tools that perform similar functions to the automation of IRC, none are ideally suited, and a custom software tool is needed.

A software tool called Restricted Interval Counterpoint Engine (RICE) is presented in Chapter 4. RICE automates the repetitive and time-consuming elements of IRC. In addition to this, it provides musically useful auditioning features for the composer.

The essential functions of RICE include:

- Searching interval space,
- Intervallic analysis of music fragments,
- Storage and management of interval Restriction Sets,
- Storage, management and auditioning of precompositional material.

Chapter 5 discusses four RICE-assisted compositions presented in the accompanying folio. The folio consists of:

- Three Songs for Soprano and Piano,
- Ten Prayers for Mixed Choir ,
- Four Fugues,
- Three Reharmonized Jazz Standards.

The musical application of IRC and the RICE software are discussed in Chapter 6. Both the method and the software are assessed to determine their successes and deficiencies in terms of musical application and utility. Considerations for modifications to both IRC and RICE are discussed.

2. IRC. INTERVAL RESTRICTION COMPOSITION

Interval Restriction Composition (IRC) is a compositional method in which the main intention is to control the interval content of portions of composed music, resulting in a distinctive harmony. The application of IRC in this research uses intervals from the equal tempered chromatic. IRC is designed to generate precompositional material and voices may be combined or used in any number up to a predetermined maximum. Concrete rhythmic values need to be applied to each voice's melodic movement, while maintaining the melodic and harmonic intervallic integrity. IRC does not provide a method a generating or controlling large-scale form; it is primarily concerned with surface-level material.

Formal examinations of the mathematics and practical considerations of IRC reveal that the method is not so straightforward to use in real life. The work demanded of the composer is extraordinarily onerous even for modest musical undertakings. Computer automation of IRC has been implemented, making the method practical.

Interval Restriction Composition requires that a composer works with a number of structural voices during precompositional stage. These voices may be manipulated during the later compositional process to obtain chords, counterpoint and various textures not present in the precompositional material.

Each discrete voice is assigned two tables of intervals. These tables determine the melodic and harmonic intervals that each voice must conform to as all the voices move together. The intervals in the melodic-interval table dictate how each voice may move from one pitch to the next. The intervals in the harmonic-interval table dictate which intervals may appear in between voices in the texture.

IRC allows complete freedom of relative motion between voices. Any number of voices may move at one time, and the intervals that form the melodic movement may rise, fall, or remain static. Harmonic intervals may be simple or compound.

Every time a single voice or group of voices move, a new set is generated. Just as the melodic movement of each voice must conform to the melodic-interval tables, so must the interval content of each new set conform to the harmonic-interval

table. The pitches in voices that move by unison are carried over into each new set.

As a starting point for generating music, an Initial Set is required, which must be a dyad or larger so harmonic intervals are present. The Initial Set serves as a point of departure for each voice. Because each voice must move by a defined interval there must be no rests in the Initial Set, since the interval between a pitch and a rest is undefined. The Initial Set need not be used in the final composition.

To generate musical material, the application of IRC is a heuristic process of moving voices by intervals in the melodic-interval tables, and then checking the resultant set against the table of harmonic intervals to ascertain that it conforms to the allowable harmonic intervals. Since the melodic interval table is used generate the new sets, there is no need to check that each voice conforms to the melodic tables. If the new set does not conform to the tables, it must be discarded and a new combination of melodic intervals applied to obtain another set, which in turn must be checked for conformation. If the set *does* conform, then it may be added to a list of Possible Sets that may follow the Initial Set.

The purpose of searching for and finding conforming sets is to allow the composer to use their discretion and choose a set that will follow the Initial Set. Once enough Possible Sets have been found to make a considered selection, one Possible Set is selected to follow the Initial Set. That new set then becomes the starting point from which the voices may move, ie, the selected Possible Set becomes the 'Initial Set', although it is no longer called that.

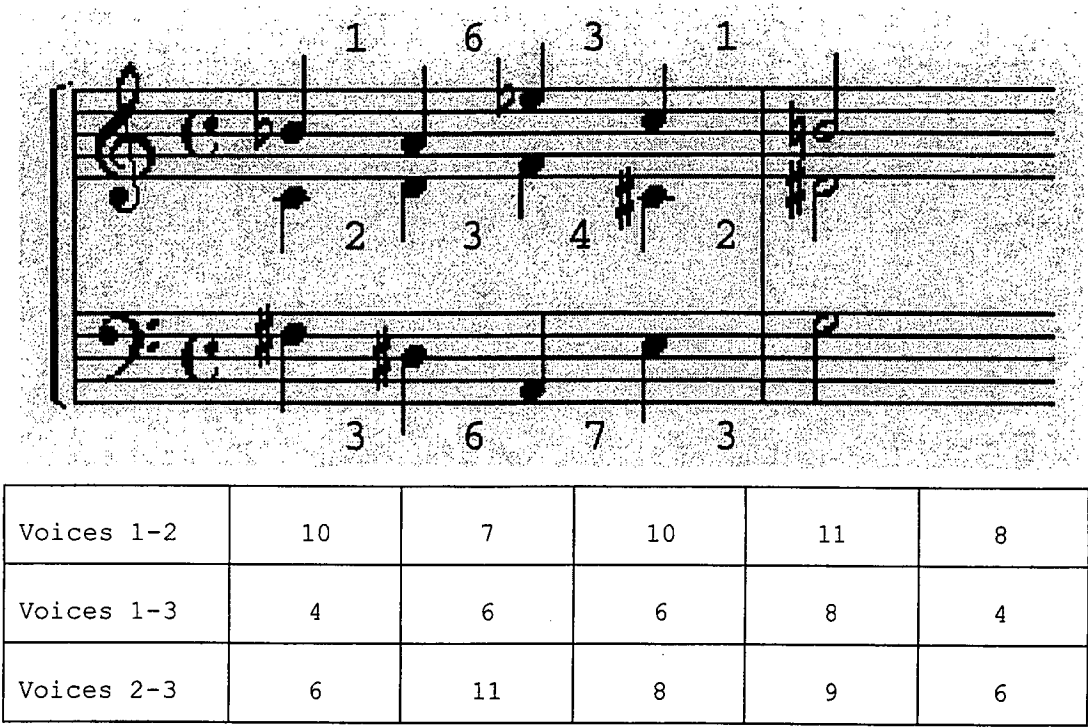
Repetition of this process of searching for and selecting one Possible Set gradually builds up a succession of sets that conform to both the melodic and harmonic interval restrictions. Each voice has an independent melodic line that is tightly integrated with every other voice in the texture. Rests are accommodated in any voice by calculating the melodic interval from the pitch just prior to the rest. Any set that contains a rest has a reduced number of harmonic intervals.

Interval tables are most easily constructed through intervallic analysis of existing music. Performing a melodic and harmonic analysis on a small fragment of music yields tables of intervals that describe the intervallic content of that fragment of

music. These tables can then be used to generate more music with the same intervallic content and hence the same sound.

In Figure 2.1, an intervallic analysis has been performed on a simple fragment for three instruments.

FIGURE 2.1 Intervallic analysis.



In Figure 2.1, the melodic intervals are marked in between adjacent pitches in each melodic line, and the intervals between each voice-pair are listed below each set. All intervals are named by semitone size, and are modulo twelve.

The Interval Restriction tables in Figure 2.2 are derived from the intervallic analysis in Figure 2.1. The intervals in each voice are collated and tabulated under the Melodic Intervals table. The intervals of each voice-pair are similarly tabulated under the Harmonic Intervals table. Interval order and repetitions of like intervals are not accounted for. For example, the two semitones ([Bb, A] and [C, B]) that appear in Voice One in Figure 2.1 are represented by the single Interval Class 1 in the Melodic Intervals table in Figure 2.2. Intervals in the tables are tabulated from lowest to highest for convenience. Intervals in the tables are not directed intervals.

FIGURE 2.2 Interval Restriction Tables

Melodic Intervals		Harmonic Intervals	
Voice 1	1, 3, 6	Voice pair 1-2	7, 8, 10, 11
Voice 2	2, 3, 4	Voice pair 1-3	4, 6, 8
Voice 3	3, 6, 7	Voice pair 2-3	6, 8, 9, 11

Once the tables of interval restrictions have been compiled, an Initial Set needs to be selected as a starting point. In this instance the last set from Figure 2.1 will form the Initial Set, [B, D#, G]. This is an appropriate choice, since the trio will continue from this point, however any other conforming three-voice set might have been used.

Figure 2.3 shows twelve different sets which have been generated from the Initial Set [B, D#, G]. These new sets were generated by moving combinations of the three-voices by intervals specified in the melodic-interval tables in Figure 2.2. Each new set was then compared against the harmonic-intervals table, and given a = if the set conformed, otherwise it was given a \neq . Sets marked with an = can be considered Possible Sets.

FIGURE 2.3 Searching for Possible Sets.

In Figure 2.3 (a) only the first voice was moved. It rose by a semitone from B to C, one of the three permissible intervals for Voice One. Voices Two and Three moved by unison and thus these pitches were simply carried over into the new set. This set, [C, D#, G] fails when compared against the harmonic-intervals table. The harmonic interval of 5 between Voices One [C] and Voice Three [G] does not appear in the list of intervals for that voice-pair, and thus this set is ruled out.

In Figure 2.3 (b), Voice One fell 3 semitones, while the other voices remained fixed. Again, the resultant set did not conform to the harmonic-intervals table, since the intervals [G#, D#] is not permitted between Voices One and Two.

The first Possible Set is found in bar 2.3 (c). The second voice has fallen a tone while the other two voices remain static. All three harmonic intervals conform to the harmonic-intervals table; interval 10 between Voices One and Two, interval 6 between Voices Two and Three, and the unchanged interval 4 between the outer voices.

In bars 2.3 (d, e, f), the upper two voices were moved simultaneously by permissible intervals, but no Possible Set was found. In 2.3 (g) however, when the two outer voices fell by a semitone and a minor third respectively, another Possible Set was uncovered with all the intervals conforming to the harmonic interval table. Bars 2.3 (h, i, k, l) all prove unsuccessful, but bar 2.3 (j) yielded a Possible Set with all three voices moving at once.

The example in Figure 2.3 contains twelve examples as this was considered enough to demonstrate the process. Many more could have been generated. Three of the sets conformed to the harmonic-intervals table, and are considered Possible Sets. The next step would be to choose just one of these three Possible Sets to become the next Initial Set. The process of searching for Possible Sets can then continue from the newly chosen set.

Only one Possible Set can be chosen to succeed a given Initial Set. Possible Sets are not connected to each other, but rather to the Initial Set from which they are generated. A list of Possible Sets is unordered, and the melodic intervals formed between successive sets in a list of Possible Sets are arbitrary. There is no more value in using a succession of Possible Sets than there is in arbitrarily connecting any sets that conform to the harmonic-intervals table, unless the composer finds them of value.

Once a satisfactory sequence of sets has been generated, this material can then be considered precompositional material for the continuation of the trio shown in Figure 2.1. Alternately, material generated by IRC could be used for any appropriate music purpose, for example improvisation, or seeding other compositional processes.

In 2.3, the twelve generated sets were not an exhaustive catalogue of all the sets. For a composer to be satisfied that the best Possible Set has been selected from all the Possible Sets, every set needs to be generated and tested, so that every Possible Set can be discovered. To be sure that every set is generated and tested, a methodical and exhaustive approach is required. Generating every set is achieved by working through two connected sets of permutations. The first layer of permutations involves applying every combination of melodic intervals for each voice. This layer is called the Interval Permutations. For each permutation of intervals, every combination of each voice's possible direction is applied to generate a set. This is the second, connected layer of permutations and they are called Direction Vectors. Each voice in the texture may move up or down by their respective intervals, or they may remain static.

Using the melodic-intervals table from Figure 2.2, Figure 2.4 tabulates the first layer of permutations which is every combination of melodic intervals using each voice.

FIGURE 2.4 Interval Permutations.

Each voice will move by the intervals specified

Voice 1	Voice 2	Voice 3
1	2	3
3	2	3
6	2	3
1	3	3
3	3	3
6	3	3
1	4	3
3	4	3
6	4	3
1	2	6
3	2	6
6	2	6
1	3	6
3	3	6
6	3	6
1	4	6
3	4	6
6	4	6
1	2	7
3	2	7
6	2	7
1	3	7
3	3	7
6	3	7
1	4	7
3	4	7
6	4	7

The leftmost column is marked Voice One and represents the interval by which Voice One will move. The middle column represents Voice Two's interval, and the rightmost column represents Voice Three's interval. Each row represents an Interval Permutation. The first Interval Permutation is 1,2,3 which means that Voice One moves by a semitone, Voice Two moves by a tone, and Voice Three moves by a minor third, the first intervals of each voice in the melodic interval table. The last Interval Permutation, 6,4,7 means each respective voice moves a tritone, a major third and perfect fifth, which are the last intervals in the melodic

intervals table. Every possible Interval Permutation is accounted for in between the first and last.

The total number of Interval Permutations is 27. The number of Interval Permutations can be calculated by the formula:

$$I_p = m_{i1} \cdot m_{i2} \cdot m_{in}$$

I_p is the resultant number of Interval Permutations, and m_{i1} represents the number of permitted melodic intervals for Voice One, m_{i2} represents the number of permitted melodic intervals for Voice 2, up to m_{in} where n represents the maximum number of voices. In this case, there are three-voices, each with three permitted intervals in the melodic intervals table, so:

$$I_p = 3 \cdot 3 \cdot 3 = 27$$

For each of the 27 Interval Permutations there is another layer of Direction Permutations which determine whether each voice moves up or down by its assigned interval, or stays static. Each of these Direction Permutations is called a Direction Vector. Figure 2.5 tabulates each of these Direction Vectors for any three-voice texture.

FIGURE 2.5 Direction Vectors.

Each voice may move up, (U), remain static, (S) or move down, (D).

Voice 1	Voice 2	Voice 3
U	U	U
S	U	U
D	U	U
U	S	U
S	S	U
D	S	U
U	D	U
S	D	U
D	D	U
U	U	S
S	U	S
D	U	S
U	SS	
S	S	S
D	S	S
U	D	S
S	D	S
D	D	S
U	U	D
S	U	D
D	U	D
U	S	D
S	S	D
D	S	D
U	D	D
S	D	D
D	D	D

There are 27 Direction Vectors in Figure 2.5. This is coincidentally the same number of Interval Permutations as in Figure 2.4. Each row represents the directions in which each respective voice will move. The number of Direction Vectors for any number of voices may be calculated using the following formula:

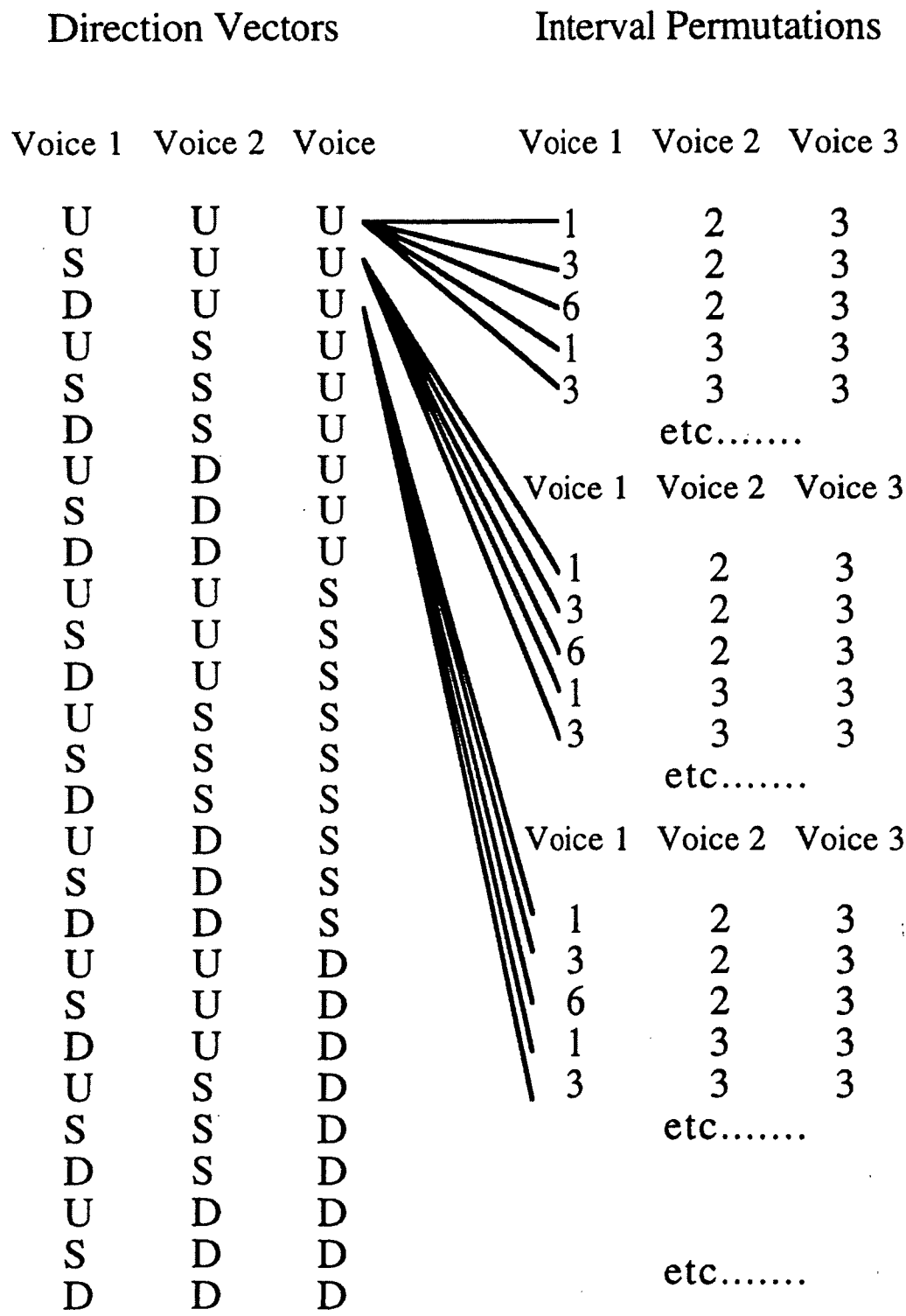
$$Dv = n^3$$

Dv represents the number of Direction Vectors, while n represents the number of voices. The exponent 3 is a constant, representing the three direction possibilities, U, S and D. So, for three-voices:

$$Dv = 3^3 = 27$$

To exhaustively generate every set, each of the 27 Direction Vectors in Figure 2.5 must be applied to each of the 27 Interval Permutations in Figure 2.4. Figure 2.6 shows Interval Permutations applied to Direction Vectors.

FIGURE 2.6 Interval Permutations applied to Direction Vectors.



The total number of sets which are generated by applying Interval Permutations to Direction Vectors in Figure 2.6 is 27*27 = 189. This can be generalised by the formula:

Ts = Ip . Dv

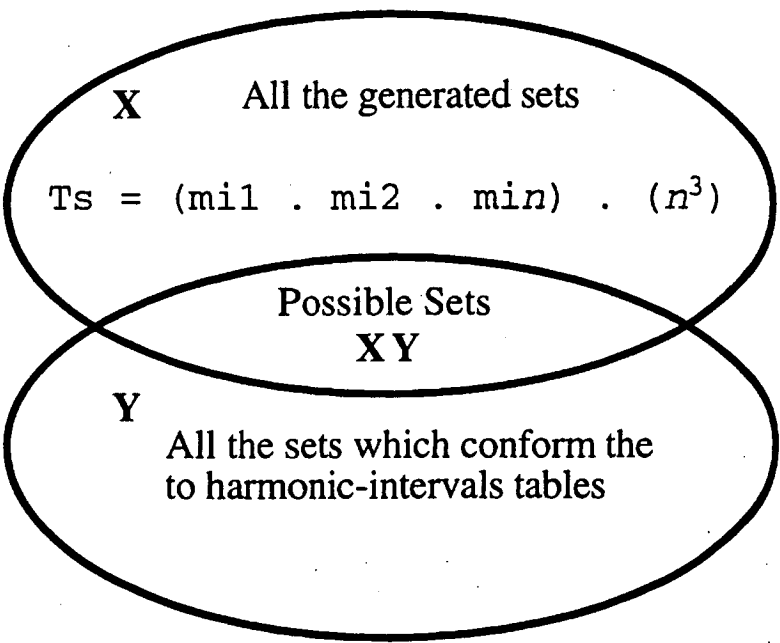
T_s represents the total number of sets, and I_p and D_v are defined from the previous formulae for calculating Interval Permutations and Direction Vectors. The full T_s formula is :

$$T_s = (mi1 \cdot mi2 \cdot min) \cdot (n^3)$$

Once all the 189 sets have been generated from the Initial Set from the trio in Figure 2.1, each must be checked against the harmonic-intervals table to ascertain which can be considered Possible Sets. The number of Possible Sets cannot be determined until each generated set has been checked. However, once all the sets have been checked, the composer may be sure that every Possible Set has been found, thus making available the widest palette of choice for the next set.

Figure 2.7 schematically illustrates the exhaustive approach to IRC.

FIGURE 2.7 Schematic representation of IRC



The domain X represents all the sets which can be generated from the Initial Set, and the domain y represents all the sets which conform to the harmonic-intervals table. The intersection xy represents the Possible Sets.

Checking the intervals of a set is straightforward for a three-voice texture, since there are only three intervals to check, ie Voices One and Two, Voices One and Three, and Voices Two and Three. However, as the number of voices increases,

the sum of the number of adjacent and imbricated intervals in a set increases in a non-linear curve according to the formula :

$$H_i = (n - 1) !$$

H_i represents the number of intervals in the set, and n represents the number of voices in the set. The number of harmonic intervals in a three-voice texture can be calculated as follows:

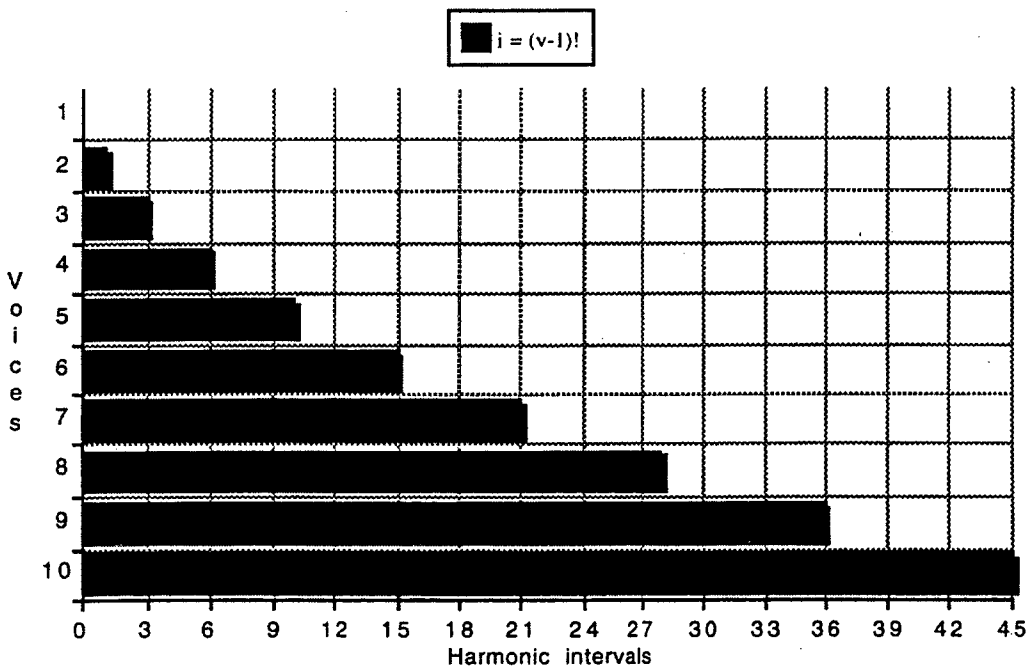
$$H_i = (3 - 1) !$$

$$H_i = 2 !$$

$$H_i = 2 + 1 + 0 = 3$$

Figure 2.8 graphically presents the number of intervals for one voice textures through to ten voice textures.

FIGURE 2.8 Harmonic intervals against number of voices.



If the composer want to uncover all possible options, the methodical and exhaustive approach discussed above must be applied. This means, however, that the work demanded of the composer is extremely high. Huge volumes of sets may be generated by the method described above. Any reasonable number of voices

may be used, and arbitrary numbers of intervals may be permitted for each voice and between each voice pair.

Application of the T_s formula reveals that these problems are exponentially compounded when music of greater complexity is considered. To exhaustively realise each set of a quartet with three intervals per voice would require just over five thousand sets to be generated.

$$T_s = (3.3.3.3) \cdot 4^3 = 5,184$$

A sextet with five melodic intervals per voice would require a staggering 3.4 million sets to be generated, checked for conformity, and all dispensed with except one.

$$T_s = (5.5.5.5.5.5) \cdot 6^3 = 3,375,000$$

Clearly this type of workload is ridiculously high, and is further compounded by the rise in the complexity of checking harmonic intervals in sets with larger numbers of voices. The H_i formula reveals that there are 15 intervals in a six-voice set. For each of the 3.4 million sets generated, up to 15 intervals would need to be checked to ascertain if each set conforms to the harmonic-intervals table.

This type of work is impossible to execute in a human time scale. Generating millions of sets by hand is a task that is best relegated to the realm of nightmares. It ignores the composer's musical skills, and reduces precompositional work to drudgery even in the simplest musical cases. Applying the processing power of computers to automate IRC is the next logical step.

3. A SURVEY OF COMPUTER ASSISTED COMPOSITION TOOLS

For the scope of this dissertation Computer Assisted Composition (CAC) tools are defined as the software tools that help composers generate precompositional or compositional material that is derived from the equal tempered chromatic. This discussion narrows towards tools that have the same features as those required for the automation of IRC.

The use of digital computers in music is such a recent and rapidly changing phenomenon that little work has been done in tracing the historical circumstances that lead to the current state. Most historical discussions focus on the hardware and software development over the last 40 years, and these discussions generally appear in an introductory style when setting the context for the discussion of a new tool or resource.

A brief historical perspective is given by Carola Bohm in *Automated Music, Interdisciplinary aspects of a musical developmental history*, (1996). Bohm points out that besides notation being the major musical achievement of the Middle Ages, the parallel codification of composition was an equally significant achievement. The ability to describe a closed system such as a compositional method also had its roots in the Renaissance. Bohm observes that the reconvergence of mathematics and music with the arrival of modern computing is an event that "stems in its most inner prerequisite from these ages and their 'automated music'".

The impact of scientific and technological thought on music that Bohm presents is supported by Albert Borgmann's, *Technology and the Character of Contemporary Life*. Borgmann discusses modern life against the backdrop of the fruits of science and technology in the form of what he calls the "Device Paradigm". This philosophically based book provides a powerful insight into the way that modern technology has reshaped our lives, and in particular the creative and thoughtful aspects of our lives. Many of Borgmann's discussions can be applied directly to considerations of the impact of technology on the creative compositional process.

David Zicarelli's *Music Technology as a Form of Parasite* ICMC (1992) is an insightful philosophical discussion of the role of contemporary technology in music, and how technology creates a need for itself just by virtue of its presence and potential. This is an important article for anyone involved in the use or development of music tools that impact on the creative process.

The majority of literature relating to CAC tools is documentary. Only a small component is independently critical of the tools available and critical of CAC as a phenomenon. MIDI, the Musical Instrument Digital Interface is often used in C.A.C. tools that use the equal tempered chromatic and although MIDI is often a lynchpin in musical applications, the frustrations of composers musicians, and programmers with the limitations of MIDI are well documented. Miller Puckette's keynote addresses *Is there life after MIDI?* ICMC (1994) is an amusing review of the impact that MIDI has had on the computer music community, painting a picture where MIDI is an instant gratification environment, and a potential technology trap. While Miller's paper deliberately is black and white to make his point, (MIDI is more useful than just a "box that spits out notes"), he correctly notes that MIDI has polarised the computer music community, and composers who use the standard must be aware of its limitations, and indeed be aware of potential prejudices from instrumentalists, other composers and computer musicians. Christopher Yavelow's paper in 1986 (ICMC), *The Impact of MIDI upon Compositional Methodology*, was written just three years after the introduction the MIDI standard. He also discusses the "way MIDI is reshaping compositional methods.", and he includes recommendations for future MIDI developments. Although the MIDI standard is designed for expansion no significant modifications have been made to date, probably due to Version 1.x's deep entrenchment.

In *Novelty, Progress and Research Method in Computer Music Composition*, (1994) Jeff Pressing discusses a list of factors that he believes contribute to the "lack of attention [in electroacoustic music] from the general public". These factors include what he calls "working methods, and the credibility and viability of compositional research".

When discussing working methods, Pressing divides traditional musical epigenesis into two main paths; "(A) Internalised hearing and development of

musical materials.", "(B) Development of materials by interaction with an instrument."

Pressing observes that if composers don't build works from internally heard cognitive maps, listeners are less likely to build or find one. He suggests that the success and value of composers music reflects their working methods.

Pressing observes that in CAC research, "... the wheel keeps being rediscovered ... to a degree not found in most fields.". He complains that articles about compositional procedures often fail to cite previous work, and often there is no attempt to integrate the current work with previous work. Pressing gives two case studies that illustrate his point.

Undoubtedly it is wasteful for research papers to replicate work that has been done previously. The rapidly changing hardware and software platforms demand that tools be redeveloped regularly, which may account for some of the apparent duplication of research work. However, in the case of software development, no two tools that are designed to archive the same result will ever be identical in function and features. Provided there is no avoidable replication, it is healthy to have a broad range of tools available to composers in order that they may be selective, matching and mixing according to their idiosyncratic needs.

Composition is a creative act. While computers can often be of great assistance in the creative process, they are not tools that intrinsically engender creativeness. Computers require strict codification of any tasks they undertake, potentially limiting creative acts to a specific domain. In *A "Pick and Mix " approach to the Integration of Composition Tools*, Eaglestone and Lunden observe that composers often are "scavengers of [amateur and professional] composition tools ... and experimenters with new and modified software.". Composers must deal with great diversity, ambiguity and volatility present in the tools available to them. They contrast this approach with that of engineering disciplines, "where professional quality tools are utilised within the bounds of 'good practice' rules and methods". While acknowledging the advantages of the structured engineering approach, they observe "the usual segregation between design objects and tools is inappropriate in a composition support environment". Eaglestone and Lunden then proceed to

describe their developmental work. Significantly, they conclude that as their research progresses,

"...an overriding requirement is to avoid prescriptive elements which restrict creativity by coercing composers into using 'good practice' tools and methods, and this method distinguishes artistic from engineering support systems."

Eaglestone and Lunden's concluding remarks describe an important philosophy in the development of CAC tools. These tools aid a creative pursuit, and vital for an artistic field's creative health is variety in inspiration and method. Good practice tools in music can encourage a 'sameness' in creative output from a variety of users. Examples of 'good practice' tools are General MIDI, MAX, C Sound, and the standard "piano-roll" sequencer design. This is because specific tools facilitate specific methods and creative process, and this is reflected accordingly in the creative output. The implicit conclusion that can be drawn from Eaglestone and Lunden's work is that variety in compositional assistance tools is desirable, if not essential to maintain diversity in musical creative endeavour.

Composers' compositional demands are idiosyncratic. There is no equivalent to the engineering "good practice" model in composition, since there are vastly differing goals, aesthetics, intentions and personalities at work in the field of composition. Therefore, it is desirable to have a collection of tools that perform similar tasks differently, all with different strengths and weaknesses, to afford composers better access to tools that actually do what they want. Composers' actions must not be defined by the tools they use, but vice-versa. A straightforward example of composers' actions being defined by the tools they use can be termed 'Mouse Music', where streams of musical events are generated by moving a computer mouse. The nature of the computer mouse defines the music.

Sever Tipei discusses the use of computer assistance in generating equal tempered intervallic precompositional material in his 1995 paper *For an Intelligent use of Computers in Music Composition*. Tipei comments that composing with computers is regarded with scepticism. He observes that the terms applied to Computer Assistance in composition need some thought. For example, the use of the term 'Computer Assisted' implies tools that can expand the composer's thinking, whereas 'Algorithmic' indicates a factory assembly line attitude toward

the use of computer in composition. Tipei observes that the "gone fishing" attitude to algorithmic composition is often praised, where the software is left to ramble and the composer "catches something of use.". Tipei sees this as an abdication of the composer's responsibilities, letting commonplace banalities take over." To take this further, the image of musically impoverished composer waiting for musical jewels to come their way by sheer luck springs to mind. Tipei also argues against tampering with the musical output to make a piece more likeable, since he regards this as asymptomatic repair of flaws, and what should be attended to is either the algorithm or the data. He regards a better option to be forcing the composer to re-evaluate the process, and that way the computer helps the composer learn more about his own ways of thinking. Tipei concludes by observing that "computers are meaningful in composition when they help expand limits".

In *An Algorithmic Approach to Composition based on Dynamic Hierarchical Assembly* Punch, Sullivan and Koehler (1991) observe that of the plethora of often technologically driven compositional tools that regularly appear, very few deal with a composition's entire problem domain, and as a result most research results in demos or toy systems which offer nothing to the composer, and contribute nothing to our understanding of the composition process. They compare this work to research that takes a general problem solving approach that meets the requirements of the domain, and note that this approach contributes to our understanding of the composition process, as well as providing software that imitates it. They outline their "generic task theme" approach, which places emphasis on first understanding the problem, then mapping that problem to appropriate underlying methodologies. Punch et al choose to define composition as "an assembly process", where the units of assembly can range from "note-flecks" to higher level assemblages, and each level can "impose constraints from above to ensure a useful 'product'.". Punch et al have underscored the importance of a composition tool reflecting the compositional domain, rather than being a tool which exists because it is technologically feasible or facile. This is critical to the success of a CAC tool, and a compositional system's domain must not be compromised by the tools that automate it. However, although Punch et al are correct in observing that many CAC tools appear as toy or demos, their conclusion that they are worthless to the composer is not correct. As discussed

above, the idiosyncratic nature of composition demands that there be many different tools available, since diversity is essential to creativity when computers are involved.

There is a large body of work that deals with attempting to codify and generate music using the tonal system. This area of work is a useful test bed for developing CAC tools because the tonal style is well documented and understood, and the success of the methods that the tools apply can easily be assessed independently of the composition system. In his paper, *Computer Analysis of Surface Details in Tonal Music*, Ficici (1991) discusses a system for analysing and identifying the function of surface-level material in tonal music. The model identifies dissonant tones, and then applies a contextual assessment to determine the function of the tone. Ficici presents an example of how the model works in practice. The work of Haig et al (1990), Steels (1986), Ebcioglu (1984) and Cope (1991) provides a cross section of tools that perform harmonisations of tonal melodies.

One of the most important features CAC software must have is the ability to search musical space. Techniques for searching are very well researched and documented in the parent field of Computer Science, and this research flows into Computer Music Programming. Some relevant searching methods, strategies and implementations are discussed.

In his 1990 paper *Music Composition as a Constraint Satisfaction Problem* at ICMC Russel Ovans "investigates the benefits of abandoning all other forms of knowledge representation (rules, logic frames etc), in favour of viewing music composition strictly as a constraint satisfaction problem (CSP)". He observes that rule based systems are frequently inefficient due to their reliance on chronological backtracking. Rule based systems typically backtrack intelligently when failure occurs, however constraint satisfaction solutions avoid failure rather than reacting when it occurs, and as a result they provide more efficient algorithms for the generation of compositions.

Ovans sites a canonical example of constraint satisfaction at work in the domain of chess, and then translates that to a simple musical example. Formally, "a CSP contains a set of variables [to which] values must be assigned according to a set of constraints." Constraints are represented as graphs. Ovans observes that CSP's are

generally solved using backtracking methods, and this method is susceptible to poor performance since it can require exponential time to solve. Ovans demonstrates how first species counterpoint may be encoded as a constraint satisfaction problem, and then makes clear that the backtracking method would be very expensive when working with four or more voices. He demonstrates that when solving all the possible counterpoints for a given cantus with backtracking, the search faces a branching factor of 25. Using constraint satisfaction the branching factor is just 5.42, with only 20 backtracks for 2746 solutions.

Assayag and Rueda (1993) discuss the "IRCAM Music Representation Group", that aims to set the "boundaries of ... music formalisation directed toward composition and analysis.". The output includes academic works, CAC tools, and a concentration of the knowledge and the musical philosophy of significant composers in the last forty years. Assayag and Rueda discuss four projects, of which the fourth is relevant here, "Partially Instantiated Musical Structure, (PIMS)". PIMS is a generalisation of structure in the functional sense, whose elements are sets and augmented with a collection of relations call Constraints. PIMS employs a hierarchical structure of sets, relations and elements, which respectively have domains, structuring and filtering behaviour. Assayag and Rueda say that "Building musical material in PIMS is a two-stage process: first constructing a suitable PIMS, and then solving the PIMS instantiation problem." Assayag and Rueda proceed to describe the algorithms for solving the instantiation. They observe that the composer generally conceives musical constraints as having degrees of importance, and proceed to discuss Soft Constraints in PIMS. Soft Constraints are described as constraints that have an importance index "between zero, (useless constraint) and one (required constraint)". Assayag and Rueda refer to Schiez (1992) for details of how these soft constraints are handled.

The searching methods discussed thus far have been deterministic, yet they need not be. Genetic Algorithms provide a very different method of searching musical space. In *Evolution of Musical Organisms*, Bruno Degazio (1996) observes that interactive computer music software often offers vast solution spaces that are daunting to explore because of their sheer size. In addition, the complexity of the interaction of available control parameters is frequently not made explicit by the

interface. Degazio's work discusses the application of Genetic Algorithms (GAs) in effectively searching unmanageably large musical spaces. He describes GAs non-specific nature, their solid theoretical background, and similarities in process to human creative activity. Degazio discusses what he sees as the simplistic application of GAs in musical projects, with a few exceptions. Degazio's work involves generating musical organisms, (which appear to be sets of MIDI events organised in a contrapuntal manner), and presenting them visually and aurally to the user for evaluation. The user then indicates the "fitness" of these organisms, and then the "fittest" organisms are bred, while the "unfit" organisms are discarded. This cycle of selection, breeding and rendering is repeated until an organism is generated which is acceptable as a musical composition. Much of the rest of the paper is concerned with the technical aspects of applying GA's to musical problems, however, he concludes by proposing further work which would apply GA's to larger scale musical elements, such as saturation, periodicity, fragmentation and varying forms of continuity.

Three software tools whose features closely overlap with the domain of Interval Restriction Composition merit discussion at this point. These tools are David Jones's *Counterpoint Assistant*, Malcolm Bell's *MAX Counterpoint Generator* and David Cope's *Composers' Underscoring Environment*,

The work of Jones (1992) describes a software tool called *Counterpoint Assistant*. Jones's tool "computes all possible realisations of a multi-voice contrapuntal texture according to user-defined palettes of allowable melodic lines and harmonies," *Counterpoint Assistant* returns all the possible solutions given a precomposed two part counterpoint, and works within the confines of academic species counterpoint. In default mode, *Counterpoint Assistant* attempts to harmonise the input with canonically varied copies of the input, but the user may configure melodic and harmonic palettes as a source of harmonisation material that are stored as pitch-class sets. The user then specifies the number of voices, which voices the original input should be mapped to and what the note offsets are, and then *Counterpoint Assistant* returns all possible harmonisations.

Jones goes on to discuss how the user has control over the Bottom, Middle and Top levels of the output, and observes significant facts about these different levels. Firstly at the Bottom Level, he notes that "writing rigorous counterpoint in

three, four, five and more voices is significantly more difficult ... than counterpoint in two voices and points out the obvious benefit of computer assistance in this circumstance. (Jones is referring to the exponential characteristics of the v_i formula presented in Chapter 2). At the Middle Level, the composer can act as director, auditioning a number of possible solutions, and then sculptor, cutting and combining contrapuntal segments. This process of screening is completely different to composing 'by hand'. Lastly in the Top Level, Jones states that he is interested in producing the best music he can, and that this involves discarding a lot of machine generated music, and modification of the output. He sees algorithmic composition (composition without direct composer participation) as simply an "unadulterated report of the efficiency of their compositional system."

Jones's justification for developing this prototype is that it allows the composer to work from a distance while obtaining results which given enough time, could have been produced by hand, allowing correct but unpredictable realisations from the given constraints.

Similar to Jones's work, the software developed by Bell (1995) *A MAX Counterpoint Generator for Simulating Stylistic Traits of Stravinsky, Bartok, and Other Composers* is a real-time interactive program that "can generate an accompanying contrapuntal line" in response to MIDI input. The generated line is dependent of the proper analysis of the style. This analysis involves building tables of melodic and harmonic intervallic frequencies and by calculating percentage weightings of the appearance of different rhythmic denominations. This analysis is required to consist of "8 rhythmic patterns and 4 harmonic pitch intervals and 4 horizontal pitch intervals.". These tables may be altered during performance. The accompanying counterpoints' pitch component is generated stochastically from the interval analysis tables, and is filtered through the pitch table. When in "free counterpoint mode", the rhythms are generated stochastically from the rhythmic denomination weightings tables. When no output is possible for a certain situation, the program will optionally ignore the horizontal interval tables, or simply insert a rest in the output.

Bell comments that frequently there were no solutions to an arbitrary input, and his solution was to relax the interval restrictions when this occurred to allow a

result to be forthcoming. This solution works due to the simplicity of writing in only two voices. Due to the exponential rise in complexity when more voices are added to a texture, a solution involving the insertion of rests or temporarily ignoring the restriction tables would not be necessarily be satisfactory in more complex music, since there may be so many rests and relaxations of the system that the system would be in danger of being compromised.

David Cope's latest work *CUE, Composers Underscoring Environment* (Cope 1996) is a mature reworking of his previous tools. Cope observes that most interactive music composition programs generate musical material by a variety of well-known processes, allowing the composer to use or discard the material as they see fit. He notes that his own previously developed tools, (Cope 1990, 1991, 1992, 1996) have "created complete works, rather than composing interactively with users.". Cope discusses the need for a compositional tool that can track a composition in progress, and combine found harmonic, motivic and structural elements with more generic elements of a composer's style to create situationally relevant music. Cope then presents Composer's Underscoring Environment (CUE); a software tool that interactively employs his well known style imitation algorithms in Experiments in Musical Intelligence, (EMI). (Cope 1991).

CUE allows composers to experimentally extend passages, and develop the potential of germ ideas and compose as much relevant music as desired in the style as evident in the previously composed music in CUE's database and in the style of the composition. CUE uses a pattern-matching algorithm to search for "signatures" (Cope 1991) in extant music, and uses a coupling algorithm that possesses "rules inheritance" for "producing highly original recompositions". Cope demonstrates with a recomposition of a Bach chorale that stylistic rules are inherited through signatures and these rules maintain integrity through recomposition, *even though these rules are not explicitly held in the program's database.*

Cope notes that "the music resulting from rules inheritance generally wanders without any sense of balance or development.". In response to this, he describes CUE's Structural Analysis Program that operates independently and quite differently from CUE's signature program. This hierarchical Structure Analysis

Program recognises structural patterns as "earmarks" which play a "critical role in CUE's ability to analyse and generate logical musical structures."

CUE's interface provides a "series of view levels", from standard notation out to "larger frames of the in-progress composition". The highest level shows the entire composition, complete with foreground, middleground and background depiction of earmarks and signatures. Cope concludes that CUE is designed to imitate, not to create anew, and comments that "highly original work music be accomplished by users."

The fact that Copes' algorithms replicate tonal music successfully without the context of tonal music is significant. Tonal music and the characteristics of the music of individual tonal composers are familiar to everyone, and success in the tonal style replication can serve as an acid test for algorithms and CAC tools. If an algorithm can arbitrarily replicate musical styles without context, (including the tonal style) the algorithm must be able to style-replicate any composer's idiosyncratic style without the context that explicit rules provide.

Cope's CUE is the result of decades of research into musical style replication and CAC and as such is the most sophisticated CAC tool related to IRC available. At the time of writing CUE is unavailable due to technical difficulties, but will appear as an adjunct to a forthcoming book titled *The Algorithmic Composer*.

The Computer Assisted Composition tools discussed have searching facilities and restrictions or constraints within which musical material is generated. These are CAC tools that offer solutions similar to that required of Interval Restriction Composition. However, they are only similar and therefore are not the correct tools for the job. A tool specifically designed to solve the problems associated with IRC is required to fully evaluate the success of this method. Chapter 4 presents the software tool called "RICE" (Restricted Interval Counterpoint Engine); software which automates and enhances Interval Restriction Composition.

4. RICE. RESTRICTED INTERVAL COUNTERPOINT ENGINE

This chapter presents a software tool devised by the author that automates Interval Restriction Composition. The software tool is called 'Restricted Interval Counterpoint Engine'.

RICE was developed in the programming language C using the Metrowerks CodeWarrior development environment for the Apple Macintosh platform. It makes use of the Graphic User Interface (GUI) for visualisation and manipulation of musical data. MIDI is used for RICE's musical input and output.

An Appendix that contains the RICE software and the accompanying manual supports this discussion of RICE. The manual describes every feature of the RICE software in detail and provides an understanding of how to make practical use of the software. This dissertation does not attempt to duplicate the function of the manual and so does not provide an introduction to the practical use of the software.

Automation of IRC has been under consideration by the author since early 1994 when it was understood that the method required exponential time to employ. The development of IRC was a response to a problem which confronts many composers, namely "How do I control the harmonic and the melodic consistently and accurately?". Controlling the sound of compositions by simply restricting where and when specific intervals may appear in, and in-between voice pairs appeared to be a very straightforward solution to the problem.

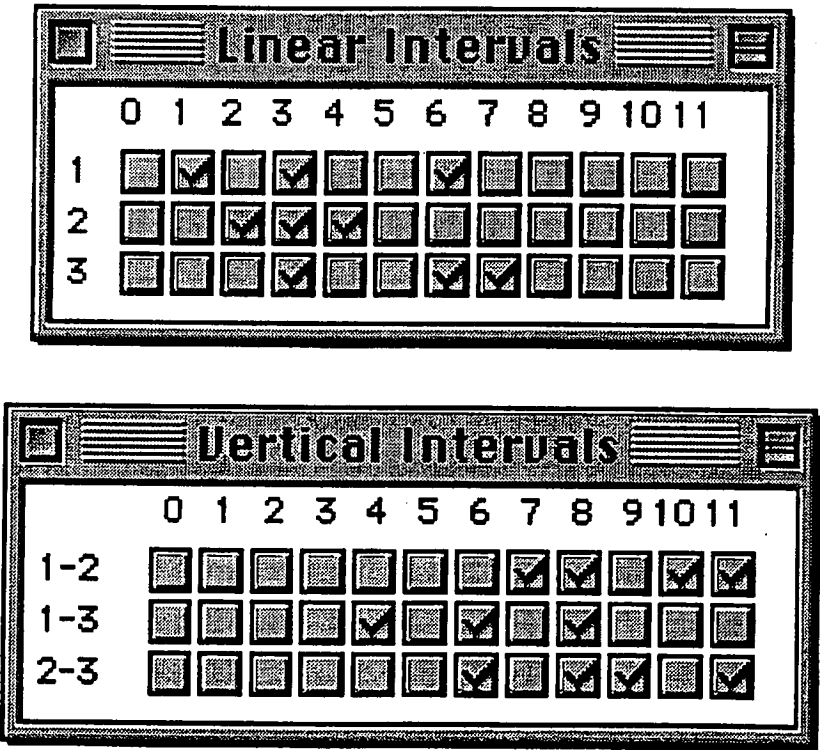
As identified in Chapter 3, a tool that automates the IRC must include searching facilities, analysis functions, and restrictions that shape the interval content of the musical material. Auditioning of musical material is also essential to IRC. In addition, facility for storing the musical material must be incorporated along with routines that allow that material to be modified in musically useful ways. The discussion covers the following nine features of RICE:

- Restrictions Tables,
- Lists of Sets formed by Independent Voices, (musical material)
- A Search Engine,
- Auditioning of musical material,
- Analysis of musical data for Restrictions Tables data,
- Manipulation of Restrictions Table data,
- Manipulation of Lists of Sets,
- Musical input and output,
- Permanent storage of works in progress.

All these features are interdependent.

Figure 4.1 shows how RICE represents tables of restricted intervals. The content of the tables are the same as the tables shown in Figure 2.2.

FIGURE 4.1 RICE's Interval Restrictions Tables.



Each table in Figure 4.1 is a grid of checkboxes. Voice numbers are listed down the left-hand side of each table, and intervals size in semitones is listed across the

top of each table. Each row tabulates all twelve intervals for each voice or voicepair, and each column contains like intervals. A check indicates that the interval is permitted in that row's voice or voice-pair. For example, the melodic intervals permitted in Voice One are 1, 3 and 6.

In addition to holding tables of melodic and harmonic intervals, RICE also provides the capacity to control the occurrence of each pitch-class in each voice. This is outside the domain of IRC described formally in Chapter 2, however it was considered a musically useful extension of the method. Figure 4.2 shows the Pitch Class Restrictions Table.

FIGURE 4.2 RICE's Pitch Class Restrictions Table.

	0	1	2	3	4	5	6	7	8	9	10	11
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Pitch Classes in Figure 4.2 are notated in numerical Pitch Class format, where [c=0]. By default, all pitch classes are permitted. RICE can work with up to eight independent voices. It is possible to alter the number of voices at any stage within RICE. When this is done, the Restrictions Tables dynamically resize to accommodate the new number of voices. Both the Melodic-Interval Restrictions Table and the pitch-class Restrictions Table resize linearly in accordance with the number of voices selected. The harmonic-interval table resizes exponentially in accordance with the $H_i = (n - 1)!$ formula presented in Figure 2.5. Figure 4.3 shows an example of the exponential resizing. The contents of the table in Figure 4.3 are derived from a RICE assisted musical analysis.

FIGURE 4.3 *An example of an exponentially resized harmonic-intervals table.*

	0	1	2	3	4	5	6	7	8	9	10	11
1-2												
1-3												
1-4												
1-5												
1-6												
2-3												
2-4												
2-5												
2-6												
3-4												
3-5												
3-6												
4-5												
4-6												
5-6												

This table is configured for six-voices, and tabulates the fifteen voice-pairs.

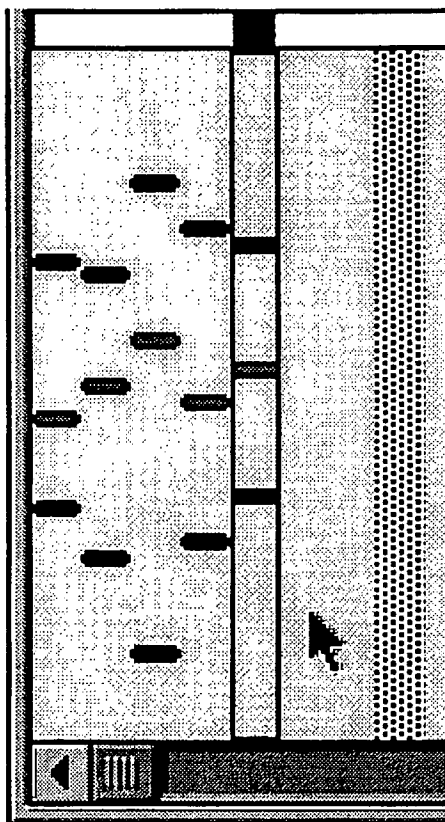
This table is configured for six-voices, and tabulates the fifteen voice-pairs.

RICE defines a collection of Restrictions Tables as a Restriction Set. Restriction Sets include the harmonic-intervals Restrictions Table, the melodic-intervals Restrictions Tables, and the pitch-class Restrictions Table. While only one Restriction Set may be active at anyone time, RICE can make multiple Restriction Sets available for activation at any time during the precompositional process.

RICE holds musical data as successions of sets made up by the movement of independent voices. RICE can work with one voice through to eight simultaneously independent voices. The range of each of the voices is defined as the 127 discrete pitches of the MIDI protocol, where Middle C = 60.

Sets in RICE are visually presented in the traditional sequencer "Piano Roll" style, where each pitch is represented by an object within a scrollable window. Figure 4.4 shows how RICE displays the trio's chord progression that was introduced in Figure 2.1.

FIGURE 4.4 *RICE's Piano Roll display for musical data.*



In Figure 4.4, each small bar in the window represents a pitch. Each of the three-voices has a different colour, and because of this colouring the contour of each voice is clearly defined. Pitches can also be coloured according to their pitchclass, and this articulates pitch-class saturation. Colouring pitches by voice and pitch-class simultaneously is also possible.

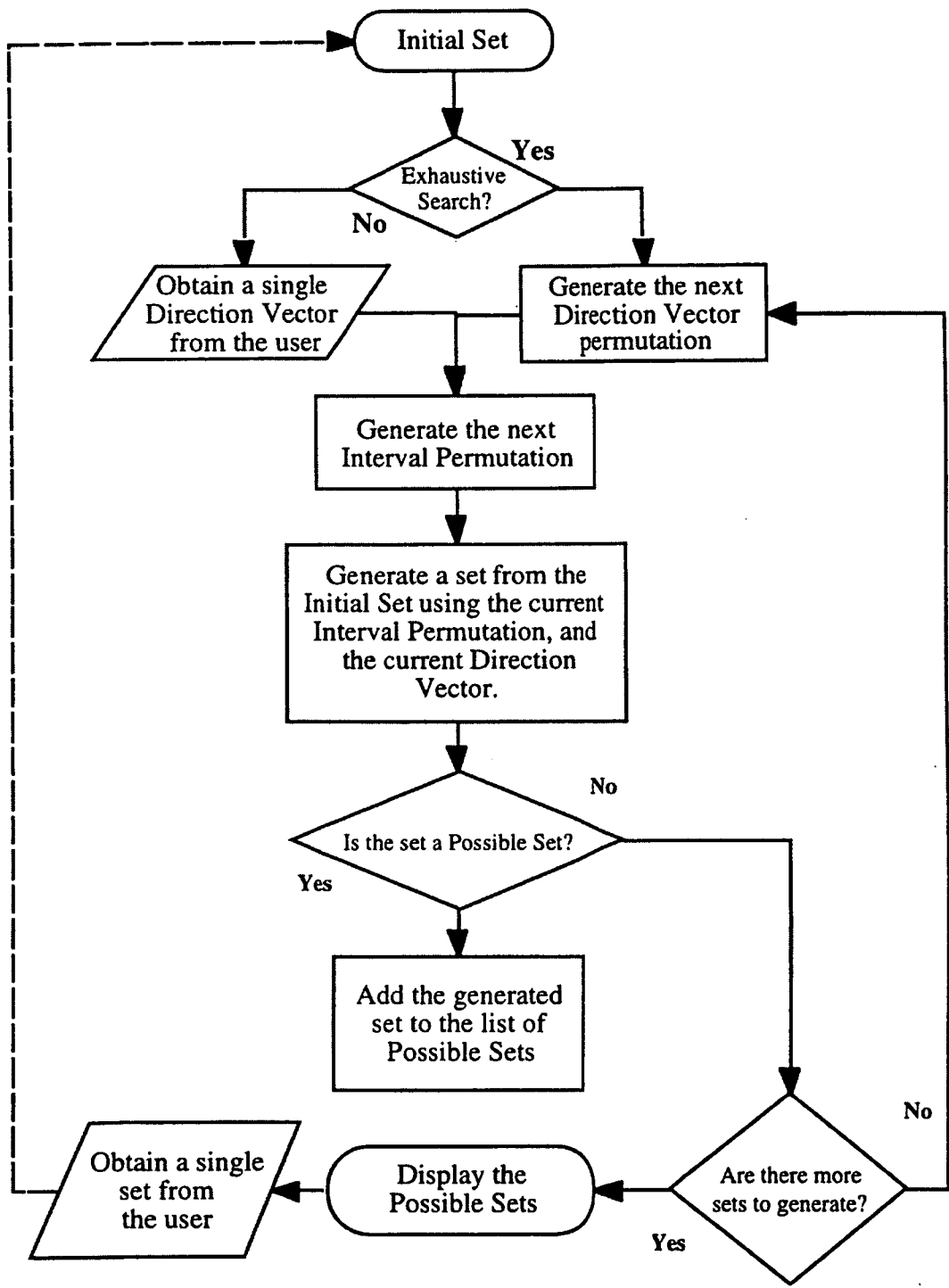
Clicking on any on any pitch with the mouse cursor will play that pitch via MIDI, and clicking anywhere in the space of a set will play the entire set. Dragging across the sets will play the sets in succession, thus allowing aural auditioning of the precompositional material in sequence. The dotted column represents the end of the file and is repositioned according to how much musical data is present.

The black rectangle in Figure 4.4 outlining the last set in the sequence is called the Centre Control, and may be repositioned by the composer. The function of the Centre Control is to indicate to RICE which set is the "Initial Set"¹. The Initial Set is used by the Search Engine to determine where searches for Possible Sets will begin.

The routine at the core of RICE's functioning is the Search Engine. The Search Engine's functioning is represented schematically in Figure 4.5.1.

¹ The RICE manual defines the "Initial Set" as the "Current Chord".

FIGURE 4.5 Flow chart for RICE's Search Engine.

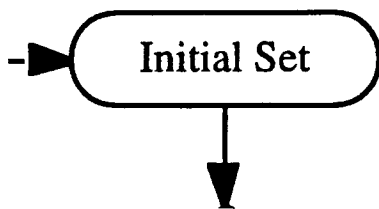


The main function of the Search Engine is to reduce the composer's workload by quickly and accurately by finding all the Possible Sets which may follow an Initial Set. This is the essential encapsulation of IRC described in Chapter 2. Figure 4.5 shows the different stages and processes that the Search Engine goes through to find all the Possible Sets for any given Initial Set. The Search Engine incorporates

filters that are not included in Figure 4.5 for the sake of clarity. Before generated sets are checked for conformity with the harmonic-interval Restrictions Table, they are checked for conformity with the pitch-class Restrictions Table, and are discarded if they fail. Sets that succeed this stage are then checked for duplicate pitch classes, ie octave and unison intervals. Sets that have duplicate pitch classes are discarded, since octaves and unisons undermine the independence of each voice, effectively reducing of number of voices.

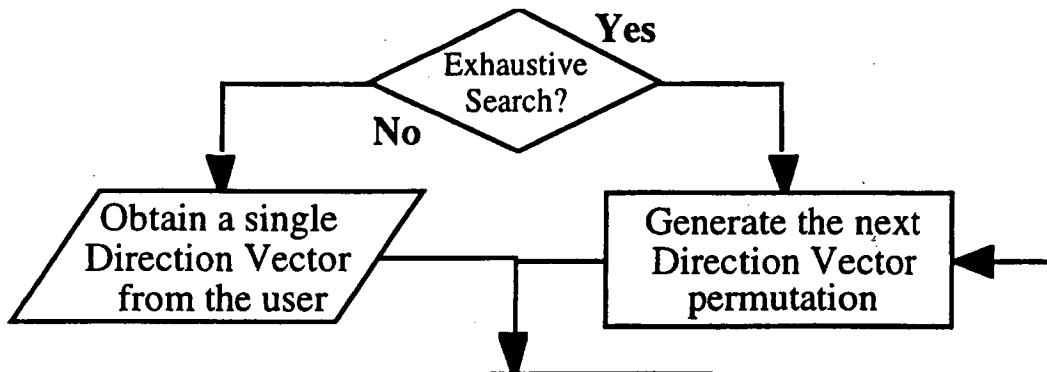
In RICE, the Initial Set is indicated by the position of the Centre Control as shown in Figure 4.4. An Initial Set is required to begin the Search Engine's processes. The rectangle with curved ends in Figure 4.5 represents the start or end of a process.

FIGURE 4.5 Extract A.



Once the Initial Set has been determined, the Search Engine is able to perform two different types of searches, Exhaustive searches, and Directional Searches. The diamond shape and a Yes/No query represent this binary choice.

FIGURE 4.5 Extract B.



An Exhaustive Search returns every Possible Set as described in Chapter 2. A Directional Search looks for a subset of Possible Sets that all conform to the same

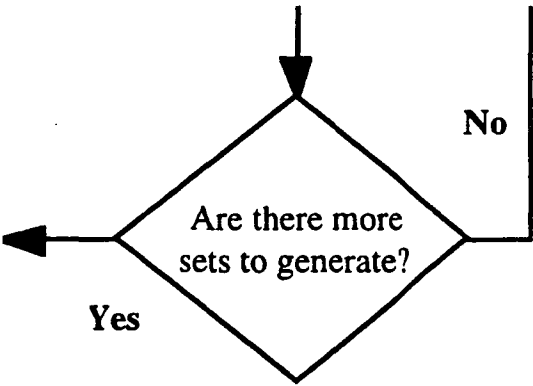
Direction Vector. An Exhaustive Search applies every possible Direction Vector to the Initial Set, while a Directional Search applies a single Direction Vector supplied by the composer. In Figure 4.5 Extract B the parallelogram indicates that there is input from the composer, while the rectangle represents a task completed by the computer.

Directional Searches are not explicitly expressed in IRCs' domain as described in Chapter 2, however they prove to be musically useful in limiting the number Possible Sets to those which are of direct interest to the composer.

The Search Engines next task in Figure 4.5 is to "generate the next Interval Permutation for the current Direction Vector". Here the Search Engine applies every Interval Permutation in turn. At this stage, only one Interval Permutation is generated, the rest will be generated when the Search Engine repeatedly loops back.

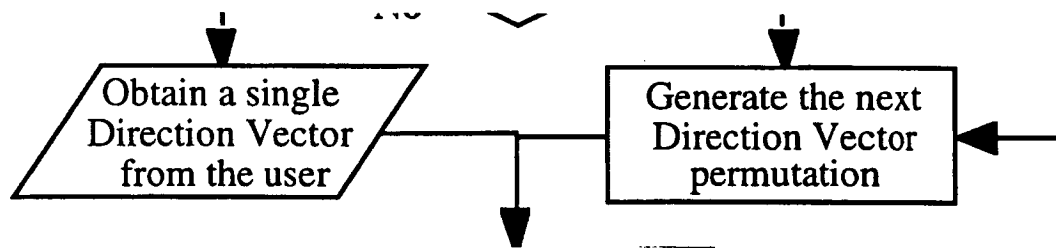
Then the Search Engine "Generates a set from the Initial Set using the current Interval Permutation and the current Direction Vector." This set is then tested against the harmonic-intervals table to check for conformity. If the set conforms, it is considered a Possible Set and is added to the list of Possible Sets. If the set does not conform to the harmonic-intervals table it is discarded. After checking the generated set for confirmation, the Search Engine then checks if there are more sets that need to generated.

FIGURE 4.5 Extract C.



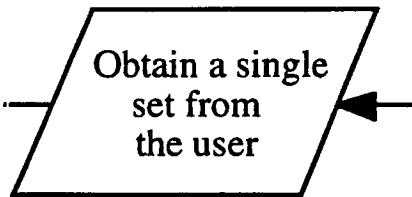
If there are, the Search Engine returns to either of the following two steps, depending on whether it has finished generating all the Interval Permutations for a current Direction Vector.

FIGURE 4.5 Extract D.



If it has not finished generating all the Interval Permutations, the Search Engine will return to "Generate the next Interval Permutation" in Figure 4.5 Extract D. Otherwise it will return to "Generate the next Direction Vector Permutation" in Figure 4.5 Extract D. If there are no more sets to generate, the Search Engine calls a routine to display the list of Possible Sets that were found.

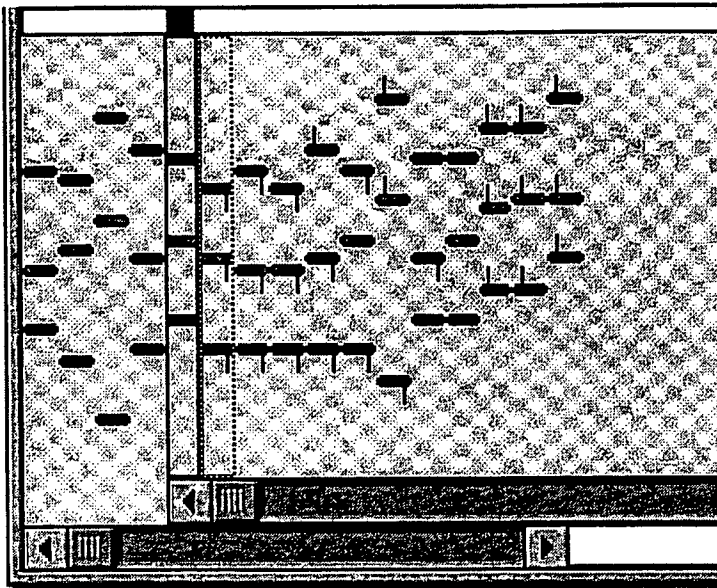
FIGURE 4.5 Extract E.



The parallelogram in Extract E represents the process of choosing a single Possible Set from the palette, and the dotted line indicates an optional return to the start of the process, with the newly chosen Possible Set to become the Initial Set.

In RICE, a list of Possible Sets is called a palette, because the composer can choose from the list. Palettes of Possible Sets are presented visually to the composer part of RICE's Piano Roll Display. Figure 4.6 shows the Palette of Possible Sets generated by an Exhaustive Search performed from the last set of the trio shown in Figure 4.4 and Figure 2.1.

FIGURE 4.6 A Palette of Possible Sets.



In RICE, sets to the right of the Centre Control are Possible Sets. Sets to the left of the Centre Control are extant sets. The set under the Centre Control is the Initial Set that the Search engine began searching from. On the colour computer screen, the background of the display window is coloured light green to remind the composer that the sets they are looking at include a Palette of Possible Sets, not just a sequence of extant sets. In contrast, the standard background colour is light grey. A second scroll bar appears under the Centre Control, allowing to composer to scroll through large numbers of Possible Sets should that be necessary.

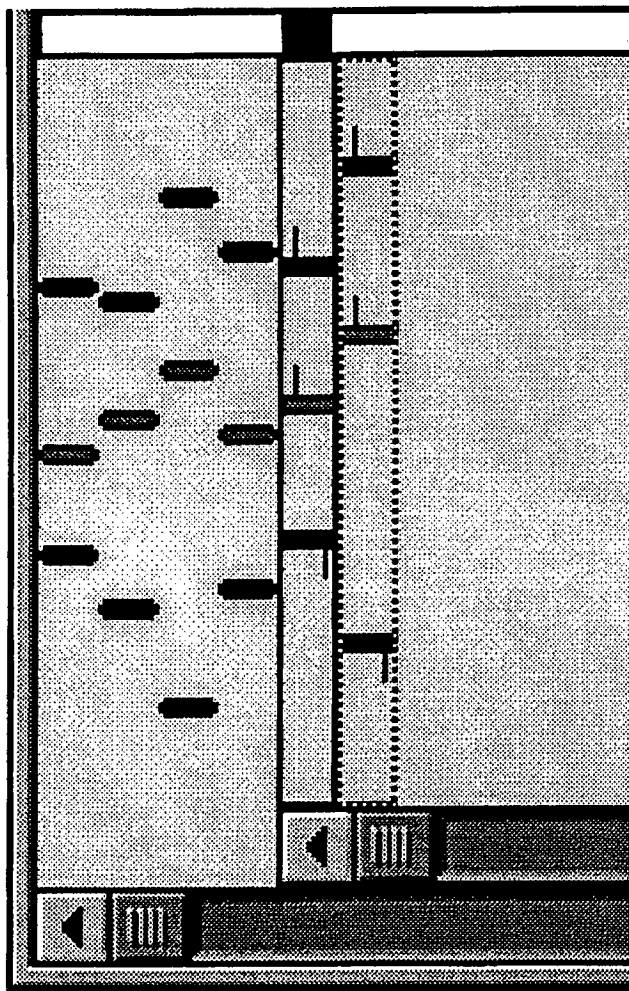
Pitches in the Palette of Possible Sets (to the right of the Centre Control) have short stems attached to them, either going up or down. These stems indicate the direction that each voice moved in to arrive at that specific Possible Set. The absence of a stem indicates that that voice did not move to form that specific particular Possible Set. The stems attached to each pitch in a set graphically represent the Direction Vector that generated that particular Possible Set.

It is important to keep in mind that sets in the Palette of Possible Sets *do not form a meaningful sequence*, unless considered so by the composer.

Figure 4.6 shows a Palette of Possible Sets that were generated by an exhaustive search. Figure 4.5 indicated that RICE can also perform Directional Searches, by

accepting a single Direction Vector as input from the composer. When this happens, a Direction Vector is applied to the Initial Set, under the Centre Control. The composer is able to attach directional "stalks"² to each pitch in the Initial Set, and these guide the Search Engine in returning only the Possible Sets that conform to this Direction Vector. Figure 4.7 shows the results of a Directional Search, using the same trio as in Figure 2.1.

FIGURE 4.7 A directional search.



The directional search shown in Figure 4.7 yielded only one Possible Set. This single possible set appears in the grey dotted Auditioning Box just to the right of the Centre Control. The Direction Vector on the single Possible Set is the same as

² The RICE manual names the graphic representation of Direction Vectors "stalks".

the Vector that was applied to the Initial Set. The single Possible Set is a subset of all the Possible Sets shown in Figure 4.6, where the same set can be found. Since there is only one Possible Set, it is the only set that can be auditioned from this search. If the composer does not want to use this set, another different search can be performed.

Because Directional Searches place severe limitations on which Possible Sets may be included in the palette, it is not uncommon for the palette to be empty. When this occurs, the composer can modify various elements of RICE to undertake a different search that yields Possible Sets. The composer may apply a different Direction Vector to the Initial Set, alter the Initial Set, or change the contents of the current Restriction Set.

The composer is required to choose one Possible Set from the palette to succeed the Initial Set. To assist in this choice, the composer is free to audition the contents of the Palette of Possible Sets. The palette may be colorised according to voice and/or pitch class, and specific sets can be auditioned via MIDI by clicking in the space of a set.

In RICE, the composer chooses a single Possible Set by placing it in the "Auditioning Box". The Auditioning Box is the grey dotted line that encircles the set directly to the right of the Initial Set under the Centre Control. The Box only appears when a Palette of Possible Sets is displayed. Because the Auditioning Box is adjacent to the Centre Control, it allows the composer to hear the selected Possible Set in the context of the Initial Set and the proceeding set progression.

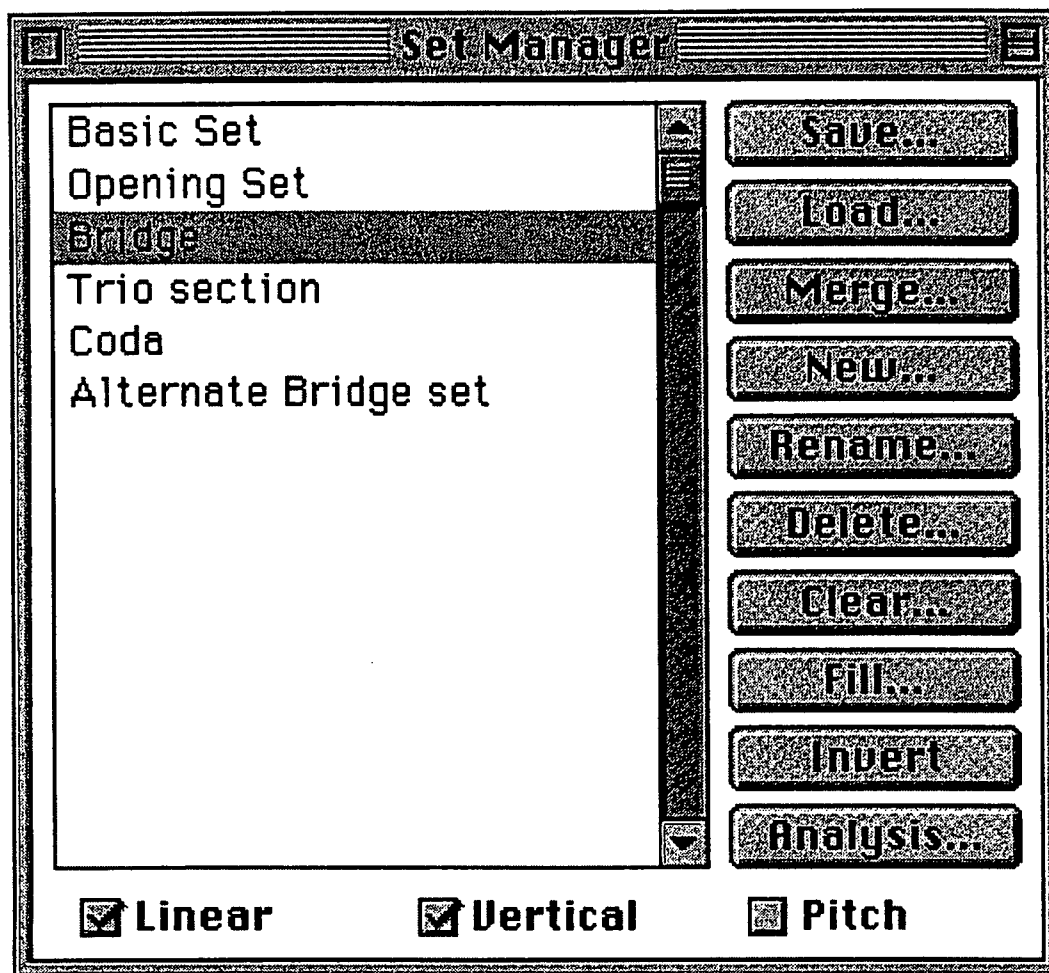
By dragging the mouse cursor from left to right, the composer can hear a set sequence that includes the Initial Set and the selected Possible Set in the Auditioning Box. If the Possible Set in the Auditioning Box is not satisfactory, it may be replaced by another Possible Set. Once a satisfactory Possible Set is placed in the Auditioning Box it may be accepted.

Accepting a Possible Set copies the set from the palette and inserts it in the set sequence after the Initial Set. Once a set has been accepted, the Palette of Possible Sets is cleared, and the Centre Control is placed over the newly accepted set. This automatic repositioning of the Centre Control allows the newly accepted set to assume the role of Initial Set for a new search.

RICE provides the capacity to analyse fragments of music for intervallic and pitch class content. This provides the composer with a powerful method of configuring the Restrictions Tables automatically. RICE is able to analyse musical data from a Type 1 MIDI file, in addition to the ability to analyse portions of musical material already present in RICE.

The RICE automated analysis of music significantly improves the chances of creating musically useful tables of restrictions. Analysing portions of music already present in RICE is particularly useful, since the composer may analyse arbitrary musical fragments that have been entered via a MIDI controller, or may reanalyse arbitrarily modified fragments of RICE-generated musical material. This can help ensure that any modifications to the RICE-generated musical material are reflected in the Restrictions Tables. RICE features a "Restriction Set Manager" which allows the composer to manipulate whole Restriction Sets. Figure 4.8 shows an example of the Restriction Set Manager Window.

FIGURE 4.8 The Restriction Set Manager Window.



A list of all the available Restriction Sets is shown at the top left of the Restriction Set Manager Window. The composer is free to select anyone Restriction Set at any time. In Figure 4.8, the Restriction Set named "Bridge" is selected.

The Restriction Set Manager provides the capacity to save the Restriction Sets independently of the other RICE data. This facility is provided to allow Restriction Sets to be loaded or merged into different RICE files thereby allowing the composer to create libraries of Restriction Sets that may be applied to any musical data.

Restriction Sets can be Created, Deleted and Renamed via the Restriction Set Manager. The contents of the each Restriction Set may be cleared, filled or inverted en masse. Clearing a Restrictions Table removes all the checks from the table, while Filling a Restrictions Table fills every checkbox with a check.

Inverting a Restrictions Table toggles the state of each checkbox. The three Checkboxes that appear at the bottom of the Restriction Set Manager Window control which Restrictions Tables in a Restriction Set will be modified by a mass action.

In Figure 4.8, the checkboxes representing the Melodic and Harmonic intervals Restrictions Tables are selected.³

Once the Restrictions Tables and the Initial Set are configured, the Working Method begins with a composer-initiated search for Possible Sets. The computer searches for and returns a Palette of Possible Sets. The composer auditions the sets in the palette, and then accepts just one set according to aesthetic criteria that are extrinsic to RICE and IRC, and intrinsic to the composer. Then the Accepted Set becomes the Initial "Set and the process is repeated. This is the Search-Audition-Accept Working Method.

The product of the cyclic Search-Audition-Accept Working Method is a succession of sets that form precompositional material. This set succession is made of *n* independent voices, which conform to the interval and pitch-class restrictions configured in RICE's current Restriction Set.

Each set in RICE represents a single rhythmic event, and each graphic bar represents an abstract rhythmic unit. For example, if a semiquaver is considered to be the rhythmic unit, then a graphic bar represents a semiquaver. Two graphic bars on the same pitch in the same voice would represent the duration of a quaver, and three graphic bars would represent the duration of a crotchet. This rhythmic unit can be chosen, changed and tied arbitrarily when the precompositional material is composed out.

Complex cross rhythms across voices can be constructed in RICE by selecting set sequences that contain oblique motion in between voices. Coupled with contrary motion, oblique motion is a key factor in defining inter-voice rhythmic complexity and independence of line. Set sequences rich in oblique motion may be composed out into complex and rhythmically independent voices. Set

³ The RICE software uses the terms Vertical and Linear

sequences without oblique motion must be composed out into homophonic textures if the intervallic integrity of IRC is to be maintained.

RICE provides the composer with a suite of editing functions that can be applied to any musical material. These include:

- Standard Cut and Paste editing functions,
- Transposition,
- Pitch locking,
- A Canon Manager,
- A 'Perform' Mode,
- Octave Box.

Any portion of musical material may be transposed in pitch space and time using easily accessible QWERTY keyboard commands. Any editing or transposition function can apply to any subset of voices. Since many of these editing functions can potentially destroy the intervallic integrity of RICE generated musical material the composer is warned to use them with care. Analysis or reanalysis of sections of edited musical data may be required to maintain intervallic integrity.

By default, pitches in RICE are freely modifiable. However this is not always desirable, and RICE provides the capacity to Lock Pitches. Locking a pitch fixes its register and temporal location relative to other Locked Pitches⁴. Whole sections of musically important data may be locked to insure against accidental modification.

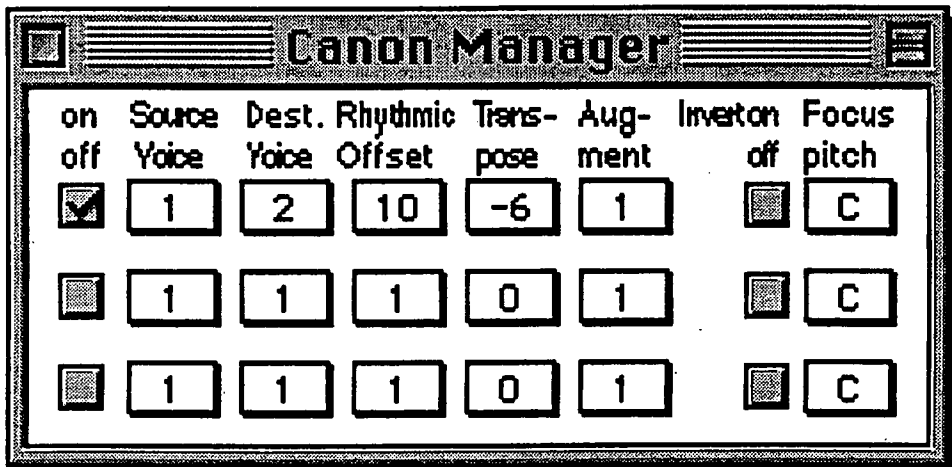
In addition to providing this basic safeguard, Locking Pitches has a musically useful application. It is possible to harmonise musical material entered into RICE using Locked Pitches. For example, a chorale melody may be entered into a single voice, and then all the pitches locked. Then, beginning from an Initial Set at the start of the melody, the standard Search-Audition-Accept Working Method can be applied. Every time the Search Engine is executed, the set directly after the Initial Set is checked to see if it contains any Locked Pitches. Any Locked Pitches are then used as a filter which is applied to the Palette of Possible Sets, and only

⁴ The RICE manual refers to Locked pitches as "Locked Notes".

Possible Sets which contain the same pitches as those which are locked are returned. Because of this feature, the chorale melody will be preserved and harmonised as the Search-Audition-Accept Working Method progresses.

A feature of RICE called the Canon Manager makes extensive use of Locked Pitches. The functioning of the Canon Manager is integrated into Search-Audition-Accept Working Method. The Canon Manager allows pitches from "Accepted" Possible Sets to be copied, placed a specific distance ahead of the Centre Control and then locked. As the Search-Audition-Accept Working Method progresses, the Locked Pitches generated by the Canon Manager become incorporated into the Search Engines' functioning, and consequently become incorporated into the precompositional material. This allows canonic style thematic unity to be embedded in the precompositional material. Figure 4.9 shows the Canon Manager configuration window.

FIGURE 4.9 The Canon Manager.



The Canon Manager allows as many canon are there are voices. Each canon may be switched on or off, and has a Source voice and a Destination voice. The Source voice specifies which voice the pitches are extracted from the accepted Possible Set. The Destination Voice is the voice in which the Locked Pitches will be placed. The Rhythmic Offset sets the number of rhythmic units that Locked Pitches are placed ahead of the Centre Control. The Canon Manager also allows canonic transformation that may be applied to the Locked Pitches generated ahead of the Centre Control. These transformations include Transposition, Augmentation and mirror Inversion about a Focus Pitch.

As discussed above, RICE allows aural auditioning of precompositional material by clicking on sets and pitches with the mouse cursor. A "Perform Mode" provides an additional method of playing back the sequence of precompositional material. Normally, RICE responds to incoming MIDI data by storing it as extant music. However, when Perform Mode is engaged incoming MIDI data is interpreted differently, and any incoming MIDI note immediately triggers the playback of a single set in the sequence of precompositional material. The precompositional material can thus be played back sequentially, with each set triggered by a single incoming MIDI note. The playback of sets matches the duration and volume of incoming MIDI pitches. The Perform Mode allows the composer to "play" the sequence of precompositional material from a MIDI controller, imposing rudimentary rhythm and dynamics. This MIDI "performance" of the precompositional material with the rhythmic information may be captured by other MIDI software, and then modified further by the composer.

RICE does not provide any direct method of ascertaining a pitches letter name other than by colour or playing it. This is a deliberate design decision which is intended to direct the composers focus toward contour, and melodic and harmonic interval. However, it is occasionally useful to gauge the relative intervals of sections of music and RICE provides a feature called the "Octave Box" to assist with this. The Octave Box feature displays a box that is always an octave high and a rhythmic unit wide. This box follows the position of the mouse cursor, and is useful for gauging whether portions of music fit within an octave. Consequently, it is useful for checking if harmonic intervals are compound or simple, and it is particularly useful in gauging if certain precompositional material will fit comfortably with a pianist's hand span.

RICE communicates musical data via the MIDI protocol. Music may be entered into RICE from a MIDI controller. Music entered this way can be played in as monophonic lines that are stored a single voice, or may be entered as a series of block sets with the appropriate number of voices. As discussed above, RICE allows the composer to audition musical data, and this is done by sending MIDI data to a MIDI capable sound device.

RICE also recognises the Standard MIDI File Format. Extant music may be loaded into RICE by importing Type 1 MIDI Files. RICE can also write Type 1 MIDI files that contain the sequence of precompositional material generated by the Search-Audition-Accept Working Method. The rhythmic unit in these RICE generated MIDI files is always a crotchet.

In addition to the MIDI file format, RICE can write the sequence of precompositional material out to a plain-text file that may be opened by a word processor. This text file contains the letter name and MIDI pitch number of each pitch in each set. Figure 4.10 shows an example of the text output.

FIGURE 4.10 An example of RICE's text file output.

VI	V2	V3	V4	V5
75 Eb	70 Bb	68 Ab	60 C	42 F#
69 A	66 F#	64 E	61 C#	43 G
68 Ab	63 Eb	59 B	58 Bb	41 F
68 Ab	65 F	60 C	58 Bb	42 F#
75 Eb	72 C	70 Bb	65 F	51 Eb

Each voice's line is tabulated in a column, and each set forms a row. This text file output is useful for composers who want to use with their precompositional material without the assistance of a computer, or for shifting RICE generated precompositional material into a software package which does not recognise the MIDI standard. RICE cannot read text files.

Printing from RICE is not possible. There is no visual data generated in RICE that is useful when out of the context of the RICE software.

RICE provides permanence through a native binary file format. This file format saves the current state of the program, including all Restriction Sets, precompositional material and any current Palette of Possible Sets.

5. MUSICAL APPLICATION OF RICE

This chapter discusses four musical works that are included in the accompanying folio. Each of the works are RICE assisted compositions. The four musical works are:

- *Three Songs for Soprano and Piano,*
- *Ten Prayers for Mixed Choir,*
- *Four Fugues,*
- *Three Reharmonized Jazz Standards.*

The role that RICE played in each works' creation is discussed, along with formal design and compositional detail. The complete scores and recordings for these works can be found in the accompanying folio.

5.1 THREE SONGS FOR SOPRANO AND PIANO

Each of the three songs are settings of complete poems by Sylvia Plath, titled *You're*, *Morning Song* and *The Night Dances*. These poems are all drawn from different phases of Plath's writing. Although they were written separately from each other there is a common thread through the three poems, in that they trace the birth of her son. Together, the three songs take around 10 minutes of performance time.

The formal structure of *You're* is simple, consisting almost completely of a single unvarying texture from start to finish. The only change in texture is the tonal climax passage at bars 67-70 discussed below. The soprano's line can be divided into three formal sections, bars 1-18, 19-52 and 53 to the end. The reason for these formal divisions is that the soprano's melodic motives cycle repeatedly, and with each repetition of the opening theme, the word 'you're' is inserted into the text to mark each of these melodic restarts.

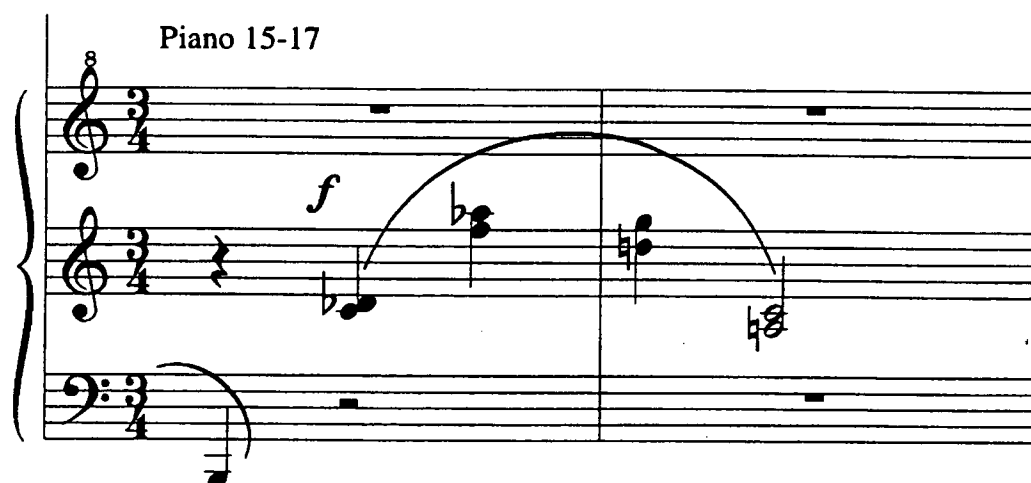
Throughout *You're*, the vocal line was written independently of the piano part. The musical material for the piano was generated in RICE, using a six-voice Restriction Set. The harmonies for the interval Restriction Sets were based on extended jazz-style harmonies.

For example, the first six notes in the piano (Bar 1) can be considered as an A^{13}/G , followed by $Abm^{13\flat7}/F$ formed by the next six notes, (Bars 2 and 3).



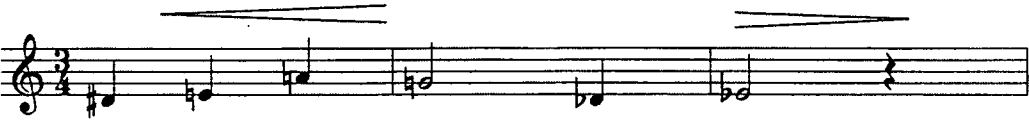
The rhythms are designed to subvert the waltz metre, and an improvisatory feel in the piano playing is intended. Although the intervallicly consistent six-voice sets are spread out registerally, they still retain their sense of cohesion and their jazzy implications because they are still discrete entities without overlapping boundaries.

The use of the RICE-generated six-voice sets in *You're* varies as the song progresses. The initial partitioning into bass, middle and upper registers is altered for the first time at bars 15-16, where an arched contour partitioning is used. This reoccurs at 22-25 and throughout the song. These six note groupings are partitioned differently, in three dyads rather than monad, tetrachord and monad in the opening of the song.



The parallel existence of unborn child and mother-to-be is represented in *You're* by two simultaneous independent musical streams. The piano and the singer perform different musical material simultaneously. The soprano sings an intuitively composed waltz-style line. This melodic line consists of thematic units that are developed as they are repeated throughout the song.

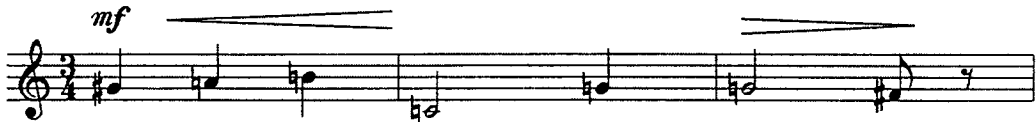
Soprano 5-7



Hap - pi - est on your hands,

Soprano 24-26

mf



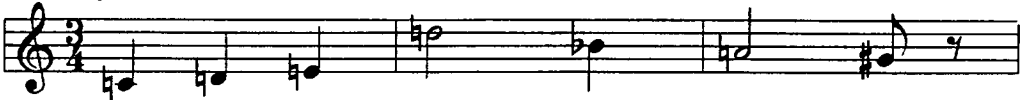
Traw - ling your dark as owls do.

Detailed description: This block contains two musical excerpts for the soprano part. The first excerpt, labeled 'Soprano 5-7', shows a melodic line in 3/4 time on a treble clef staff. The notes are G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4 (quarter), A4 (quarter), and G4 (quarter). The lyrics 'Hap - pi - est on your hands,' are written below. The second excerpt, labeled 'Soprano 24-26', is marked *mf* and shows a melodic line in 3/4 time on a treble clef staff. The notes are G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4 (quarter), A4 (quarter), and G4 (quarter). The lyrics 'Traw - ling your dark as owls do.' are written below.

The vocal line works against the syllabic pattern in the poem, with the melodic material supporting the phrases of meaning in the text.


You're makes frequent use of word painting. For example, the relatively wide vocal interval of a minor 7th is used in bars 8-10 and 35-36 to symbolise height.

Soprano 8-10



Feet to the stars, and moon - skulled,

Soprano 35-36



O high ri - ser

Detailed description: This block contains two musical excerpts for the soprano part. The first excerpt, labeled 'Soprano 8-10', shows a melodic line in 3/4 time on a treble clef staff. The notes are G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4 (quarter), A4 (quarter), and G4 (quarter). The lyrics 'Feet to the stars, and moon - skulled,' are written below. The second excerpt, labeled 'Soprano 35-36', shows a melodic line in 3/4 time on a treble clef staff. The notes are G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4 (quarter), A4 (quarter), and G4 (quarter). The lyrics 'O high ri - ser' are written below.

The reference in the text to Australia is represented in Bars 43-46 by a metrically displaced waltz rendition of the opening of Australia's national anthem, *Advance*

Australia Fair. This rendition is entirely unsupported by the piano, which helps render the magnitude of the distance suggested, ie, "further off than *Australia!*".

Soprano 43-46
with pomposity

Far - ther off than Aus - tra - li - a

At bars 60-62, the predominant pattern of one-syllable-per-pitch in the vocal line is temporarily subverted by a melismatic setting of the word ripple.

Soprano 60-62

rip - ples

Although the piano does not sympathetically support the vocal melodic line, it does respond to the meanings in the text, extending the word painting process. In bar 28, the piano supports the text by its absence, it is 'mute as a turnip' for one whole bar, the only tacit bar in the piano for the whole song.

Bars 27-29

mf *f*

Mute as a tur - nip from the

In bar 36, the apparently incongruous high piano trill represents an oven timer ringing. The text at this point, 'O high riser, my little loaf' is making a reference to

the colloquial expression 'she has a bun in the oven' - meaning a woman is pregnant.

Bars 35-36

O high ri - ser

tr

Bars 63-66 provide a 'jumpy' lead into the climax of *You're* through the use of disjunct intervals. This climax (bars 67-69) is a tonal melody supported by a conventional tonal progression in the piano that is theoretically 'right, like a well done sum'. The progression at the climax can be described as :

ii6 (V65) V7 (V65) vi

Since this progression contains a deceptive cadence, tonal convention demands that the music continues, and the last five bars provide a final cadence for the song. The previous musical style begins anew for conclusion at bar 70 with the text ‘a clean slate’.

Bars 67-70

f *piu mosso* *ff*

Right, like a well done sum. A clean Slate.

f *f*

ii6 (V65) V7 (V65) vi

This is the only place in *You're* where the piano works with the vocal line, providing a chorale-style accompaniment to the vocal line.

The final cadence in the piano leaves the listener in an expectant mood designed to heighten the anticipation of the beginning of the *Morning Song*, which has a dramatically different texture.

The basic piano material throughout *Morning Song* is a series of five-voice RICE generated sets, with different interval content in the different formal sections which are detailed below. The vocal line is composed to extrinsically to RICE to complement the interval content of the piano, and weaves in and out of the pitches found in the piano chords.

In *Morning Song* the piano and voice work together in a sympathetic manner, reflecting a supportive environment. The textural basis of the song is homophonic. The soprano's line supports the emotional content of the text, and unlike in *You're* the piano underpins the soprano's musical line.

Morning Song divided into four sections with the structure, AB A' C shown in Figure 5.1

FIGURE 5.1 Formal divisions in *Morning Song*.

Bars 1 -26	A	Wonder of the arrival of a new baby.
Bars 27-54	B	Contemplation of becoming a mother.
Bars 55-64	A'	Response to a babies cry.
Bars 65-80	C	Ethereal reflection upon a babies crying at morning twilight.

The different sections are defined by the use of different RICE sets, alternating between dissonant and diatonic intervals for each subsequent section. Consequently, the harmonic worlds are distinctly different. These formal sections are also defined musically by thematic, tempi and registral differences.

The first section (A) is characterised by melody and accompaniment, with brief piano interludes between lines and stanzas of the poem. *Morning Song* 'set going' with no preamble, and the opening theme is immediately taken up and 'set going' in the piano in the second bar and throughout the A sections, while the vocal line in bar 2 sings 'fat' punctuating crotchets with the regularity of a ticking watch.

Bars 1-3

$\text{♩} = 90$

Love set you go - ing like a fat gold watch

pp

The thematic material from the soprano's opening line is taken up by the piano at bar 3, and developed at bars 11-14 and throughout the song.

Piano 3-4

mp

Piano 11-14

Bars 7-8 are a mini 'babies cry' climax that is developed fully in bars 60-63.

Soprano 6-8

3 3

foot soles And your bald cry

Soprano 60-63

3 3 3

night gown Your mouth o - pen clean as a cat's

The vocal line in bar 15 is 'echoed' directly in the top voice of the piano in bar 16.

Bars 14-17

mp

Our Voi - ces e - cho mag - ni - fy - ing Your ar - ri - val

p mp

The piano crotchets between bars 26 and 27 (sections A and B) build a high 'wall' between the two formal sections.

Bars 26-28

mp *mf* *Slower*

as walls I am

mp

The B section (Bars 27-54) quickly fills out the previously unused lower register of the piano, and employs RICE generated diatonic intervallic content in a rhythmically static accompaniment figure which is decorated by an offbeat countermelody of crotchets above the vocal line. This change in rhythm and intervallic content supports the change of tone in the text.

Bars 29-33

more your moth - er Than the cloud That dis - tils a mi - rror

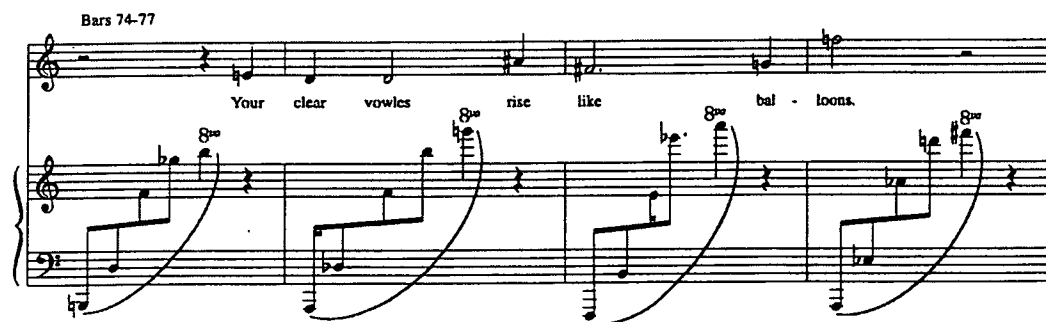
mf *mp*

This B section is divided into two subsections, and from Bar 41 the accompaniment's rhythmic pattern is quickened by a ratio of 2:3, reinforcing the text's depiction of the mild anxiety involved in waking and listening for the cry of a sleeping baby. At bars 47-50, the piano prepares an expectant 'listening' chord, and remains quiet while the singer describes the sound heard.



From bar 52 the piano moves to the return of A' which breaks the quieter musical landscape with a baby's cry, and leads to the climax of *Morning Song* at bar 62, prepared in the first A section.

The very different nature of the coda style C-section reflects the ethereal, beautiful and uncluttered depiction of a baby's cry at morning twilight. The image of 'sounds rising as balloons' underscores the piano accompaniment figure, where the rising ostinati is buffeted gently by cyclic irregularities in the rhythm.



As with *You're*, the cadence here is at textural extremes with the opening of the following and final song, *The Night Dances*.

Formally, *The Night Dances* can be divided as shown in Figure 5.2

FIGURE 5.2 Formal divisions in *The Night Dances*.

Bars 1 -22	A	Represents a child's dances.
Bars 22-34	B	Motherly reflection.
Bars 34-49	A'	... developing towards ...
Bars 50-53	C	Climax.
Bars 54-71	D	Dark, and brooding.
Bars 71-90	A'' merged with D	Coda, thematically thinning and unwinding.

As in *Morning Song* these divisions represent the use of markedly different RICE Restrictions Sets, shaping the harmony and melodic shapes of the music. The cross over of thematic and intervallic content is more sophisticated here, with the material from A, C and D fusing in the coda.

The Night Dances is the chronologically longest of the suite and draws on thematic ideas from both *You're* and *Morning Song*. However, unlike the previous two songs, the vocal line is tightly integrated with the piano chords. The movement from musical independence between the piano and the vocal line in *You're* to musical interdependence across the parts in *The Night Dances* represents the bonding that occurs between mother and child after childbirth. However, in *The Night Dances* there is still a degree of independence in the two streams, since the vocal line is in a duple metre and the piano is predominantly triple. The two streams however, share the same pulse, and the tetrachords that are broken up between the piano and the vocal line progress at the same rate.

Bars 1-3 ♩ = 90 - 100

f

A smile fell on the grass.

The Night Dances uses four-voice RICE-generated sets, and the pitches of the vocal line are extracted from the piano chords at a rate of one per tetrachord, leaving a trichord in the piano for each note in the vocal line. The integrity of RICE generated chords pervades most of this song, as practically all of the composition was undertaken in RICE.

The manner in which the tetrachords are partitioned between the voice and the piano in section A, of *The Night Dances* (bars 1-22), gives the music a feeling forced animation. The vocal line is largely restricted to multiples of crotches and crotchet rests, and this has been done to force the trichords into specific durations in the piano line. In a sense, the vocal line is forcing the piano's rhythmic activity. The feeling of energy forced out is similar to a baby's experimental movements with its body, energetic but with very little fine motor control. The bursts of

accented triplet chords in Bar 3 and throughout the song appear without the vocal line are echoes of the surprising and unrefined movements that babies make. Section B (bars 23-34) sees a return to the more ethereal, contemplative B section from the previous song. The feeling is, however, not positive.

Bars 23-27 *mp*

I shall not en - tie - rly sit emp - tied of beau - ties the

Static chords are appoggiated way up above the melody line providing accompaniment, and the vocal line is freely composed above the full tetrachords in the piano. The bridging bar at 35 provides a taste of a new thematic idea developed fully in Section D (bars 54- 71). On the return, the A' section (bars 35-49) is more rhythmically complex and varied.

Bars 35-38 *f*

Their flesh bears no re - la - tion.

The Climax, section C (bars 50-53) takes the triplet crotchet theme in the piano and punctuates the vocal line.

Bars 50-53 *f*

The co - mets Have such a space to cross.

This abuts directly with the D section (bars 55- 71), where the piano is again working with tetrachords, and the vocal line is sympathetically composed above.

Bars 54-57

mp *mf*

Such cold - ness, for - get - ful - ness. So

The conclusion of this section is in the extreme low register of the piano, against the text 'the black amnesia's of heaven', bars 68- 70.

Bars 68-70

mp *mp*

The black am - ne - sias of heav - en.

After a bridging bar , the material from the A section returns at Bar 71, however this time it becomes broken up and juxtaposed with themes from the D section.

Bars 76-79

mf *mp*

Fal - ling like bles - sings like flakes Six

The last 12 bars see an unwinding of energy and thematic complexity which takes its lead from the text. The material in the coda is drawn exclusively from the RICE generated tetrachords.

5.2 TEN PRAYERS FOR MIXED CHOIR.

Ten Prayers for Mixed Choir is a collection of choral settings of selected texts from Michael Leunig's Common Prayer Collection, (1993). The titles of the selected prayers and the required voices are shown below in Figure 5.3.

FIGURE 5.3 Voicing of Ten Prayers for Mixed Choir.

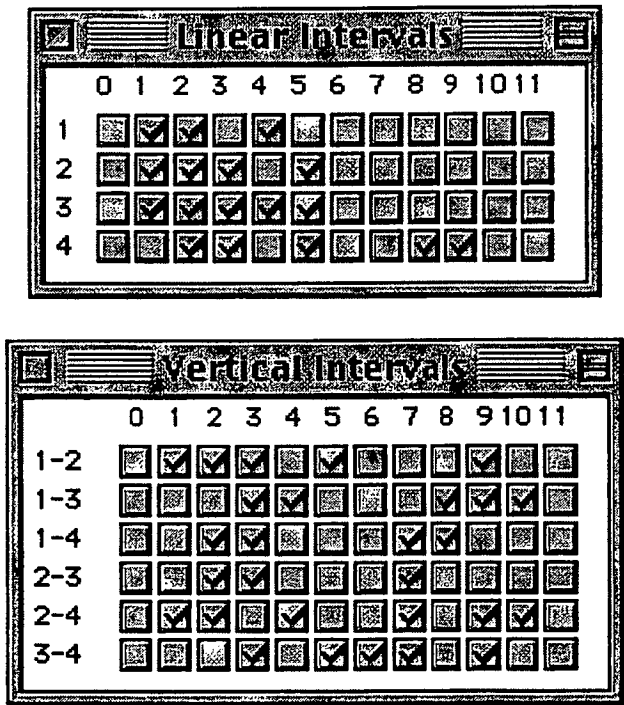
I	<i>When the heart</i>	SATB
II	<i>God help us to live slowly</i>	SATB
III	<i>God bless the lost</i>	SATB
IV	<i>God bless this tiny little boat</i>	SATTB
V	<i>God help us if our world should grow dark</i>	SSS
VI	<i>Love one another</i>	SAATTB
VII	<i>God be among</i>	SATB
IIX	<i>God rest us</i>	SSA
IX	<i>Let it go</i>	TB
X	<i>Dear God</i>	SATB

Each of the prayer texts is short, and the length of the musical settings varies from around 20 seconds to a couple of minutes. *Ten Prayers* take about ten minutes of performance time.

In the published text of Common Prayer Collection, each prayer is accompanied by an illustration, and these illustrations influenced the musical character of each setting. The mood of each illustration played an equivalent role in the shaping of the music, as did the text of the prayers themselves. The full meaning of the text can only be understood with the illustration and so the illustrations are included with the text in the score.

Much of the compositional work for these prayers was undertaken in the RICE software. Melodic fragments were harmonised in RICE, and phrase lengths and phrase contour were composed in conjunction with the text within RICE. The main compositional work done outside RICE was shaping the phrases with dynamics and rhythm. The source of intervallic material used to seed the precompositional material generated in RICE included polyphonic tonal works, such as Bach chorales and fugues, and jazz harmonies. For example, the sections of the prayer *God Bless this tiny little boat* use harmonies configured in the Restriction Sets shown in Figure 5.4. The restriction sets were configured from Bach's various harmonisations of *O haupt voll Blut und Wunden*.

FIGURE 5.4. A Restriction Set used in *God Bless this tiny little boat*.



A variety of compositional devices were used for these choral settings. Two of the prayers, *God Bless the lost* and *Love one another* were composed without RICE, and are not discussed here. For five of the eight RICE-assisted musical settings, chords with tonal implications were used, variously drawn from the choral, fugal and jazz idioms. The tonal chords were used without strong regard for their implicit tonal function, yet the affect of these chords' sounds was taken into account. In *When the heart, God bless this tiny little boat*, and *Dear God*, dissonant tonal chords are used to colour specific words in the text, either drawing

connections between words, or heightening tension of words and phrases. Movement to consonant chords resolves the tonal dissonance of these chords, but not by following theoretical tonal rules of harmony and voice leading. Often these dissonant chords are the result RICE analysis of passing tones as voices move in the original music. These chords are heard only fleetingly in the original music, but are now brought to the fore and assigned a significant role in these musical settings. The jazz style settings do not follow the classic [ii V I] cycles of most jazz. Essentially, the expectations that the tonal chords set up are subverted.

The intention of the musical settings was to present an augmented version of the text which reflected the illustration. Comprehension of the text by the audience without program notes was a very high priority, (reinforcing the notion Leunig's common prayers), and as a result much of the writing is in rhythmic unison, taking advantage of the unity that choral forces offer. Independence of melodic shape, in each voice within the predominantly unison rhythmic framework was also considered important. Consistency of intervallic sound was considered essential to give each prayer a distinctive character, and as a result the intervallic integrity inherited from RICE was unaltered.

In the accompanying folio, a solo soprano and a string ensemble perform a complete recording of *Ten Prayers*. A separate score for the string performance is included. A choral score is also included, and performances of the choral version of *Ten Prayers* have been performed, but recordings are unavailable at the time of submission.

Since each of the *Ten Prayers* are miniature, they are discussed individually.

When the heart is a text setting whose intervallic content is drawn directly from phrases in the Bach Fugue XXII from Book I of the *Well Tempered Clavier*. The tenor forms the dux with a pitch canon that is replicated six pitches later (Bar 3) in both the soprano and the alto, depending on voice register. This pitch canon persists for half of the prayer.

Bars 1-5

The musical score for Bars 1-5 shows four voices: Soprano (S), Alto (A), Tenor (T), and Bass (B). The lyrics are: "When the heart is cut or cracked or broken Do not". The score includes dynamic markings (*mf*, *mp*, *f*) and phrasing slurs. The Tenor part is the dux, and the Soprano and Alto parts enter with a pitch canon six pitches later.

The alto sings a "simple song" (the first phrase of Mary had a little lamb) in bars 24-25 and the soprano leads the prayer to a cadence.

Alto Bars 24-25

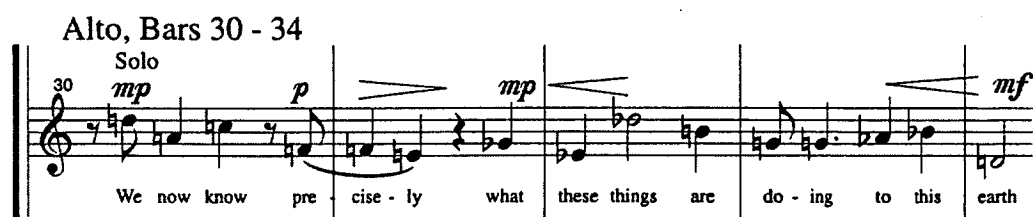
The musical score for Alto Bars 24-25 shows the Alto part singing the lyrics: "sing A simple song". The score includes a dynamic marking (*mp*) and a phrasing slur.

The intervallic material used in *God help us to live slowly* is sourced from the opening phrase of this prayer, which was composed extrinsically to RICE and then entered into RICE and analysed. It is the first prayer to use the tritone cadence Amen, which appears in eight of the ten prayer settings.

God bless this tiny little boat is a chorale harmonisation. The opening phrase of the chorale melody *O haupt voll Blut und Wunden* is used in the soprano, and the bass line from Bach's harmonisation No.74, (Bach 1941) of this chorale is used in the bass. These two phrases set the first stanza of the prayer. The second stanza has the same structure, and is set in a similar manner as the first, transposed down a semitone for a sinking affect.

God help us if our world should grow dark is the first prayer to break from the SATB setting. The five-voice intervallic material is not drawn from extant music, but constructed specifically for this prayer.

The first section of the relatively long prayer *God be among us* is spoken by an alto soloist. This section of the text sets the context for the actual prayer that follows in the musical setting. The musical setting is intended to be forceful, using six independent voices and a strong accented chanting style. These relatively large forces represent the world, and the soloist's passage from bar 30 represents the individual who cannot hide from responsibility of their role in environmental care. The solo is set using the complete pitch-class aggregate; there is nothing hidden.



God rest us is the only setting in the *Ten Prayers* that is strongly polyphonic. The texture for *God rest us* and is modelled in the style of the long flowing contoured lines of Renaissance polyphony. The soprano leads with the chorale melody *Jesu meine Freude*. The melody is augmented, and rhythmically simplified.



Each line of text is set within a phrase, with the four-voices starting and finishing together, however inside each phrase, the voices demonstrate rhythmic independence.

Let it go uses a five-voice jazz harmony, and is intended to be sung in a tight jazz a cappella style. The jazz style is inspired by the illustration. This setting exhibits more polyphony than many of the other settings, however phrases still begin and conclude together. The Amen is added as a quirky cadence - there is no Amen in the original text.

Dear God is made of freely composed phrases, which match the ebb and flow of the text. For example in bar 13 the spoken rhythm for the text "eternity" is augmented, where as the text "it nestles there" matches a natural speaking rhythm.



The intervallic material is a fusion of jazz style harmonies and the harmonies from Bach chorales.

5.3 FOUR FUGUES

As a reaction to the predominantly homophonic *Ten Prayers*, *Four Fugues* were written with the intention of using RICE to assist in composing a collection of explicitly polyphonic works. Even though the work is titled *Four Fugues*, only the last fugue shows the full complexity which deserves the title Fugue. The first three may be considered Fugettass, as they are more canonic in style. The first three fugues are in five voices, and the last is in four voices. The fugues were written as abstract music in the style of J.S. Bach's *The Art of Fugue*, and since the fugues were written as abstract music they could be arranged (for example) for double piano, or organ, string quintet with Double Bass or larger ensemble. A double piano score of *Four Fugues* is included in the folio, along with a recording.

The orchestration of these fugues is for a diverse collection of instruments that were available as an ensemble at the time of writing. No recording of that ensemble's performance is available. The instrumentation is flute Bb and bass clarinet, soprano and baritone sax, (also doubling tenor), trumpet, tuba, cello and piano. In the orchestration the instruments were paired off in dissimilar timbres and like registers. With four instrumental pairings and the piano, there were five groups to take the five-voices of the fugues. The last fugue has only four voices, and so the piano was omitted from the orchestration. The instrumental pairings are shown in Figure 5.4. The variation in pairings and the variation of baritone sax to tenor sax was done to accommodate registral extremes in the fugal lines.

FIGURE 5.4 Instrumentation for orchestration of *Four Fugues*.

<i>Fugues 1-III.</i>	Bass Clarinet and Tuba, Cello and Baritone Sax, Piano, Soprano Sax and Flute, Clarinet and Trumpet
<i>Fugue IV.</i>	Bass Clarinet and Tuba, Cello and Tenor Sax, Clarinet and Flute, Soprano Sax and Trumpet,

Each fugal voice is partitioned between its instrument parings.

Fugue I, Bar 1

The image shows a musical score for the first bar of Fugue I. It consists of three staves: Soprano Sax, Tenor Sax, and Trumpet. The Soprano Sax staff has a melodic line starting with a half note, followed by a quarter note, and then a half note, marked with a dynamic of *mf*. The Tenor Sax staff is empty. The Trumpet staff has a melodic line starting with a half note, followed by a quarter note, and then a half note, also marked with a dynamic of *mf*. The key signature has one flat (B-flat), and the time signature is common time (C).

The themes and each countersubjects' partitioning remained constant across the various different instrument pairs except in the piano. The piano is reserved for the final subject entry in each fugue where the timbrel complexity was at its densest, and the piano's line is doubled in octaves to help it cut through the welter of solo instruments.

A roughly even mix of compositional work was done intrinsically and extrinsically to RICE. Like in the composition of the *Ten Prayers*, the desire to maintain the intervallic integrity inherited from RICE was strong. Only one set of intervals is used for each fugue, giving each fugue a distinct intervallic 'sound' character.

For each fugue, a suitable theme was devised which would support contrapuntal development. These themes are variously modified themes from fugues from

Bach's *Well Tempered Clavier*. Three of the themes use relatively longer rhythmic durations than the countersubjects, so there is a sense of independence between subject and countersubject, and so that the theme will stand out among the contrapuntal texture.

The theme for *Fugue I* was composed by the author.

Theme from *Fugue I*



The theme for *Fugue II* is modified from the theme of the F Minor Fugue from Bach's *Well Tempered Clavier*, Book I.

Theme from *Fugue II*



Fugue III's theme is modified from the theme from Bach's G Minor Fugue from the *Well Tempered Clavier*, Book I.

Theme from *Fugue III*



The four bar theme for *Fugue IV* is modified from the theme of F# minor fugue from Bach's *Well Tempered Clavier*, Book I.

Theme from *Fugue IV*



Each fugues' theme was entered into RICE and harmonised in the maximum number of voices for that fugue. In essence, this meant that all the countersubjects were composed simultaneously. Care was taken to ensure that the resultant voices

would be rhythmically independent, and that their contours were as contrasting and complementary as possible. Once each theme was harmonised in five (or four) voices, each countersubject was rhythmically and melodically assessed for its strength and independence. The most successful countersubject became the first countersubject, which accompanied the theme after the theme had appeared on its own at the start of the fugue. The subsequent countersubjects were added incrementally, until the full complement of voices were employed, resulting in the original full voice harmonisation of the theme.

Since each countersubject appeared in each register after each voices' entry, (except in *Fugue II*), they had to be composed with a limited register to keep voice crossing to a minimum. The countersubjects' registral placement in each voice was changed when the fugues were composed out as good registral voicing could not always be guaranteed. Because of this problem, a number of potential harmonisations were discarded. Sometimes, simply changing the order of the countersubjects appearance solved these registral voicing problems. Each subject entry appears in octave transpositions only. Because these are atonal fugues, it was considered that no function could be derived from transposition of the subjects. In addition, transposition to levels other than the octave would also have compromised the intervallic integrity of the fugues.

The first three fugues are literally palindromic, and the final fugue has palindromic characteristics. During the retrograde reflections, the middle two fugues include chromatic inversion. This exploits the well-known fact that intervallic integrity is maintained under chromatic inversion and retrogression. As a result, one block of RICE generated countersubjects can sustain these transformations. Dynamics and articulations in the orchestration are mirrored under retrogression.

Fugue I's retrogression pivot is at barline 24-25 and the reversed countersubjects disappear one by one until the theme sounds on its own to conclude the fugue. There is no chromatic inversion.

Fugue II differs from the other three fugues since the theme and the countersubjects do not move through the different instrumental pairings. The countersubjects simply appear against the theme and prior countersubjects in their

designated instrumental pair. This static timbral and registral distribution is compensated for with a chromatic inversion in the retrogression on the way out of the fugue. The central pivot of the fugue appears at barline 18-19. Either side of this main pivot are two smaller palindromes, which are incorporated, in the larger palindrome. The last countersubject is completed at the end of bar 15, and the last countersubjects entry from bar 13-15 is pivoted about barline 15-16, adding a further three bars before the central pivot, barline 18-19. This small last - countersubject palindrome is repeated on the way out as part of the larger palindrome, in bars 19-24. The chromatic inversion involves all instruments, however the piano is the only instrument to retain its countersubject. The theme in bars 1-18 was partitioned between the Flute and the Soprano Sax. In the second half of the fugue, the Bass Clarinet and Tuba take the theme, at the extreme low register. The inner two instrument pairings also swap countersubjects under inversion. The register of the fugal lines is also inverted, since the chromatic inversion is about a fixed pitch Middle C rather than simply a pitch class. The arbitrary choice of middle C as a pivot of inversions was chosen because it gave good registral placement of the voices under inversion.

The form is similar to *Fugue II*, but the piano takes the fourth subject group before the pivot. The main pivot of retrogression and inversion is at barline 14-15. There is a four bar palindrome, two bars each side of barline 10-11, mirroring back the last subject group as in *Fugue II*. In bars 13-14, bars 9-10 (the last countersubject) appears inverted, with the theme in the Flute and the Clarinet. Then from bar 1526, a literal retrogression and inversion of the first half of the fugue ensues. The mini palindrome and inverted material is incorporated.

Fugue IV is the most sophisticated of the four fugues. It employs palindromic and inversive techniques, but the form is not literally symmetrical, rather more organic. It has two subject groups, including a stretto and a new group of countersubjects with free material. The fugue also cadences with an instrumental tutti, a dramatically different conclusion to the previous fugues, which died away in the manner in which they grew.

Fugue IV is in four-voices, and as a result the piano is tacit, and the four lines are partitioned through a instrumental grouping varied from the previous fugues. Instead of an inversion or a retrogression after the last countersubjects entry, a

second subject group begins at bar 17. It takes the form of a stretto, with the bass instruments introducing an inverted form of the theme only two bars later. The countersubjects that make up this subject group are different - but unified by the intervallic integrity inherited from RICE. Barline 28-29 is the pivot of a palindromic mirroring of the second subject group, but this only continues literally for one sounding of the four bar theme. At bars 36-37 the second subject group cadences, and a very tight stretto of the countersubjects occurs, leading toward a climax and an all-instrument cadence at bar 41.

5.4 THREE REHARMONIZED JAZZ STANDARDS

These three short works are intended to explore a different application of the RICE software. Three standard jazz tunes were entered into RICE and harmonised using intervallic content from a specially prepared sequence of five-voice jazz harmonies.

The three tunes are:

- *All the things you are* Hammerstein and Kern
- *Here's that Rainy Day* Van Heusen and Burke
- *Watermelon Man* Herbie Hancock

Essentially, harmonisations of the lead melodies were undertaken in RICE. The melodies were entered into RICE, locked and harmonised as detailed in Chapter 4. The resulting harmonies were then voiced as piano accompaniment and bass supporting the lead melody. The same Restriction Set was used to harmonise all three tunes, giving them all the same harmonic flavour, however, the rhythmic characteristics of all three tunes are different. Although this intervallic material uses tonal chords, none of the tonal implications that these chords carry are followed through, giving the music a sense of constant freshness and subversion.

All the things you are is set in a classic jazz style with walking bass, 'comping piano and lead melody ready for embellishment. The extended 36 bar (32+4) ABCA' form has been maintained with the only repetition of harmony occurring at the beginning of the A' section.

Here's that Rainy Day is traditionally a ballad, but has been voiced here as a bossa-nova, with a straight eight melody, appoggiated piano style, and a rhythmic

Latin style bass, alternately accentuating the downbeats and the backbeats. Its ABAB, form has been harmonised without repeats.

Watermelon Man retains its jazz-rock fusion style, with a melodic and rhythmically punctuating bass, and electric piano accompaniment. Unlike the other two tunes, modifications have been made to the melody. The harmonisation does not make use of functional tonality, and since the melody depends of the tonal 16 bar form to provide harmonic interest under the repeating main motive, the melody has been altered slightly to provide that interest, while retaining the 16 bar form. In the original harmonisation the main theme, (bars 1-4) is repeated, supported by a IV chord. During the repeat in this harmonisation, the freshness of a IV chord under the repeated main theme is replaced by a transposition of the theme by a semitone. Similarly the second theme which appears in bar 9 is repeated up a semitone, and then back at the original level, providing a sense of movement away and back that is achieved with the original tonal chords, V - IV - V - IV - V in the original harmonisation.

6 DISCUSSION AND SUMMARY

The primary demonstrable success of Interval Restriction Composition is its capacity to provide the composer with consistent control over the surface-level intervallic sound of a composition. This level of control was the goal that led to the method's derivation.

An unanticipated product of the application of IRC was a phenomenon that Jones (1992) describes as "Unpredictable realisations of the given constraints." Since the IRC restriction tables only record the presence of intervals as opposed to their position or frequency, an atypical or unimportant set found during an interval analysis will be given equal weighting as common, typical sets. For example, dissonant passing tones in tonal music play an unimportant harmonic role, yet if those harmonies are given prominence, the character of the harmonic 'sound' changes. This means that palettes of Possible Sets will have the same interval content as the original section analysed, but the frequency of specific sets will be quite unlike the original. A further unanticipated product of the nature of IRC's restrictions tables is that sets which did not appear in the original analysis can legitimately appear in the pallet of Possible Sets, while still retaining the original interval integrity. This is because the restrictions tables simply record the presence of an interval in the analysis, not the presence of an interval in a specific set. It is worth noting that both these unanticipated manifestations of the IRC system arise from the interval representation format of IRC, not a stochastic process.

RICE was able to render IRC practical to employ which in turn allowed IRC to be tested for musical utility. In addition to the basic automation of IRC, RICE proved successful in providing enhancements to IRC. These enhancements include visual and aural auditioning which allow the composer to work with precompositional material on an intuitive level. There is no need for the composer to have knowledge of the interval or pitch class content of the material at hand when the Search-Audition-Accept Working Method is applied. The composer is free to choose and respond to sounds and shapes while being assured that the quality of musical material being generated is consistent. Because the Search Engine yields Palettes of Possible Sets quickly and deterministically, the composer need not be

especially precious with any material generated and accepted. Because the composer is removed from the specific pitch class and interval content of the material at hand, the restrictions that typically come with that knowledge are absent. In essence, the composer works in abstract, and their aural and visual (contour) preferences are magnified since these are the primary guiding forces at work.

RICE makes available specific functions that are facilitated by the computer codification of the IRC domain. These functions would not be appropriate or possible outside the computer environment. Locked Notes are a specific example of this kind of feature, as they were not within the original IRC domain. RICE's method of harmonising melodies using Locked Notes is sensible only in the computer assisted environment, since composers working by hand would devise other means to achieve the same end. The inclusion of features such as RICE's Canon Manager demonstrates a compositional method expanding in ways not possible without the computer, since the Canon Manager relies on a computer implementation of Locked Notes to function. RICE's Perform Mode is another example of a computer dependent extension of the compositional method, as is the intuitive auditioning of musical material. This does not invalidate the utility of a composer's pianistic or instrumental abilities for auditioning the musical material, but rather complements and augments these abilities.

IRC has a narrow compositional domain since it demands the composer work with a fixed number of voices. Like many other precompositional methods, IRC pays no regard to other musical parameters such as timbre and harmonic rhythm. These elements are left until the "composing out" stage of the compositional process.

Although IRC is discussed in terms of the equal tempered system, other temperaments could be used. The RICE automation of IRC requires that the composer work within the equal tempered chromatic, use modulo twelve representations of interval and pitch class and work in no more than eight voices. These are instances of restrictions imposed by the RICE software tool rather than IRC. RICE's implementation of IRC allows only for interval analysis and generation of intervals that are modulo twelve. This means that melodic intervals generated by the search engine will always be less than an octave. To

accommodate melodic and harmonic intervals greater than an octave, RICE allows for quick and easy octave transpositions of pitches. These transpositions do not interrupt the intervallic integrity because of octave equivalence.

Since IRC generates only surface-level material, there is no part of the method's domain that deals with the large-scale structural development of a musical work. The primary problem with this approach is that there is a risk that the resultant music will be aimless, Cope (1996). Clearly when using IRC or any method, the composer must take into account the formal design of the composition as a separate task. RICE does not assist with the design of large-scale form, however it can assist with other smaller scale design aspects. Because of the auditioning facilities of RICE, the composer can consider the middle-ground development of musical material generated. For example, phrase length, contour, intervallic content development, climax points and smaller structural divisions can all be auditioned from within RICE. In this instance, RICE provides a partial solution to a deficiency in IRC.

RICE does not provide any explicit rhythmic control of the musical material, and this lack of balance in IRC's domain is emphasised in its automation. The abstract rhythmic units discussed in Chapter 5 can make the composition of precompositional material generated in RICE problematic. Using oblique motion to control polyphony is not an intuitive or familiar method of working with rhythm. This is not to say that rhythmic complexity and polyphony cannot be achieved with RICE generated material, however as it stands IRC tends to impose a bias towards homophony, and abstract rhythmic units restrict the ease with which rhythmic complexity can be achieved. These biases are comparable to the pitch class and interval biases imposed by instruments, learnt patterns and manuscript. A suite of tools for the analysis of rhythm and local calibration of RICE's abstract rhythmic units could be a valuable addition to RICE's features if the output is to be notated directly.

One significant problem with IRC is the simplicity with which it stores intervallic information in the restriction tables. The main difficulty is when performing analysis in RICE, it is very easy to saturate the restriction tables to a point where practically every interval is permitted, and so no specific intervallic 'sound' is apparent. This generally occurs when musical samples larger than about eight sets

are analysed. Of course, the interval content of the musical sample is crucial in determining whether the restrictions tables will be saturated. For example, analysing music that is derived from an all interval 12-tone series will saturate the restrictions tables very quickly. The simplest solution is to only analyse small fragments of music to obtain quality restrictions tables, and this is what was done in composing the music in the accompanying folio. However, more drastic changes, such as revisions of how IRC and RICE store analytical abstractions could lead to deeper musical analysis and higher level generation of musical material. Possibilities include, but are not limited to, modifying IRC to identify whole sets and lines rather than just the interval content, or establishing a chain of rules about how musical elements move, Cope (1996). These chords and rules could be replicated in searches, resulting in material presented to the composer for auditioning. However, as IRC's unpredictable results from simple restrictions tables demonstrates, each model would have to be carefully studied to determine its behaviour before being implemented in software.

Both RICE and IRC impact on the compositional process. The requirement that independent voices be used strongly influences the generation of the precompositional material, even though the composer may treat the material as block chords that may be voiced as a texture. This is done in *You're*, in *Three songs for Soprano and Piano*. In addition, the absence of explicit rhythmic thought in both RICE and IRC can impact strongly on the rhythmic characteristics of the final composition.

The ability to control the sound of musical material had a strong impact on the composition of the four works in the folio. RICE's Restriction Sets may be modified or exchanged at any time. This can lead to sharp delineations between sections of musical material generated by different Restriction Sets. These differences can be clearly heard between different musical settings in *Ten Prayers for Mixed Choir*, and different formal sections within songs from *Three Songs Soprano and Piano*. In some instances, RICE Restriction Sets can be swapped for every alternate set that is generated. When this is done, the resultant progression of sets demonstrates a patterned merging of the restrictions from the Restriction Sets that are being swapped. Many of the Prayers were written using this swapping technique. At the other extreme, the *Three Reharmonized Jazz*

Standards were all harmonised using the one Restriction Set, and as a result they all share the same intervallic sound which binds the three tunes together. This is typical of the small harmonic palette that jazz music uses. Each of the *Four Fugues* used a different single Restriction Set and so they all sound different, even though the same formal compositional process was applied to each piece to yield fugue-style forms.

IRC defines only one kind of search, and this impacts on the way that sequences of precompositional material are generated. RICE's Search-Audition-Accept Working Method incrementally adds material after a single set, and the process is entirely melodic. Other search types are possible, and they would impact differently on the generation of precompositional material. For example, if searches were conducted to find sets which fitted in between two extant sets rather than following an extant set, then precompositional material could be generated to fill in or 'backfill' between predetermined structural goal sets. Another possibility is that searches could return palettes of possible progressions rather than Palettes of Possible Sets, and this would impact on the compositional process by further removing the composer from mechanics of the musical material. This mayor may not be advantageous. Divorcing the composer from the materials may mean they loose touch with the practical 'performability' of the material they are working with. Many variations on search types can be imagined, and these would all impact on the compositional process in different ways and each would require further study.

Clearly, RICE's intuitive Search-Audition-Accept Working Method has an impact on the sound of pieces composed with RICE assistance. The composer can be sure of the aural reality of each set and individual voice at the precompositional stage. RICE distances the composer from the mechanics of the material and its generation, and this is reflected in the freshness of many of the choices in line and harmony in RICE assisted compositions. Many of the sets used in the four works in the supporting folio have tonal derivations, but as discussed in Chapter 5, do not follow through the tonal implications. This is because the composer is able to audition any of the available options free of the voice leading and function implications imposed by the functional tonal system. Any functional tonal progress ions that arise from use of tonal chords in RICE are a product of the

composer's ear. When non-tonal harmonies and lines are used in RICE, choices are still made free from any system other than the interval and pitch class content of the Restriction Sets. The composer's ear can be the creative force.

There are two further significant impacts that RICE itself has on different aspects of the compositional process. The first is the length of time taken for searches to be completed. This of course is dependent to some degree on the processing capacity of the computer being used, however the ramifications of the T_s formula show that the explosion in complexity easily outruns the incremental advances in processing speed of new generations of computer hardware. As a result, it can take up to ten minutes for moderately complex searches to be completed on moderately powerful computers. Search times of this magnitude impact on the compositional process by breaking up the composer's concentration. Typically with searches of this length, the composer is demanded to think musically for about a minute during the Audition-Accept phases of the Working Method. Then, nothing is demanded of the composer for (up to) ten minutes while a search is conducted. This allows for a lot of socialising, coffee drinking, newspaper reading and window gazing to be legitimately incorporated into the compositional process, which is not entirely a bad thing, and in this context, faster search methods and faster computers may be a dubious advantage. The main problem with slow searches is that they engender a discontinuity of musical thought with which the composer must deal.

The second significant impact that RICE has on the compositional process is the exclusion of arbitrary precompositional decisions. The codification of tasks that software tools demand to be able to carry out tasks tends to preclude arbitrary creative actions, Zicarelli (1992), Borgmann (1984). This is true of any computer software and applies to musical CAC tools. While RICE can accommodate arbitrary modifications of the interval content of precompositional material through reanalysis, these changes must be within the bounds of RICE's compositional domain. For example, RICE does not easily accommodate changes in the number of voices, and any changes to interval content must be equal-tempered, discrete pitched, modulo-twelve and fixed. More flexible software design can alleviate many of these musical problems providing the underlying compositional domain is sound and well understood.

Undoubtedly, both IRC and RICE require the incorporation of explicit methods for controlling the rhythmic content of musical material. The most straightforward way of introducing rhythmic control would be to explicitly codify the abstract rhythmic units into the RICE software, and provide a suite of tools to visualise and manage rhythmic durations. Ideally such a software feature would help minimise the rhythmic biases of RICE and IRC discussed earlier in this chapter. Explicit control over the rhythmic content of musical material would allow larger portions of the compositional process to be undertaken in the RICE tool itself, simplifying the compositional process to allow the composer to focus on the musical issues at hand.

Neither RICE nor IRC explicitly support large-scale development and structuring of musical material. While this may not be a problem, there is certainly scope for development of a feature in RICE that assists with the development of formal structure of musical material. One possibility is to introduce a backfill search method (discussed above), that allows the space between structural goals to be filled in. The introduction of large scale contour management, and large scale tracking of the use of Restriction Sets could also prove musically useful in assisting the composer with the construction of large scale form. To be a truly able to work with larger scale musical elements, both IRC and RICE would need to be capable of analysing, storing and employing information about large scale musical elements. This requirement of Computer Assisted Composition is becoming apparent, and is at the forefront of research, Cope (1996).

Useful technical improvements that could be made to the RICE software include voice crossing filters, voice range checking, and routines to check the suitability of a line for a specific instrument. User definable colour systems for representing pitch would greatly enhance the musical utility of the program. One composer complained that the use of colour in RICE prevented him from using the program because he was colour blind. During the audition stage of the Search-Audition-Accept Working Method, the composer has to visually and aurally audition a relatively large number of Possible Sets, typically in the order of tens. The ability to sort the Possible Sets within a palette according to musically useful criteria would be a significant advantage. By default, Possible Sets are presented in an order which reflects the pattern in which RICE generates the Direction Vectors,

from down, through static to up. Being able to automatically sort the Possible Sets according to pitch, interval content, register, or motion type (oblique, contrary, similar or parallel) would assist the composer in temporarily filtering out unwanted types of Possible Sets. This could lead to better choices from the palette, and consequently better musical material.

Improvement in the speed of RICE's Search Engine are possible through the application of Ovans' "Constraint Propagation" (1990). The dynamics of constraint propagation mean that exponential improvements in efficiency can be achieved in the face of exponential complexity. Consequently, the ramifications of the application of constraint propagation techniques to the RICE's Search Engine could be significant. For example, faster searches mean that more searches can be conducted in the same clock-time, and the possibility of returning Palettes of Possible Progressions rather than Possible Sets could be a viable reality. Possible Progression can be thought of as paths of Possible Sets that can be auditioned as a sequence. Any sequence may be accepted to succeed the Initial Set, thus allowing for a faster and more integrated working method.

On occasions, a RICE search will return no Possible Sets. There are two viable possibilities for assisting the composer in this situation. The first possibility is to help ensure that the situation does not arise by allowing Soft Constraints to be incorporated into the Search Engine, Assayag and Rueda (1993). The second possibility is to provide the composer with graphic feedback about why the search failed to return any Possible Sets. Specific restrictions could be pinpointed as the key to a search's failure, and the composer could intelligently modify the restrictions according to this information.

Interval Restriction Composition and RICE are demonstrably musically useful tools. Since the method and its automation have proved musically practical and useful thus far, further investigation of both IRC and its automation would appear fruitful and valuable.

REFERENCES

- Ackernan, P. (1994). *Design and Implementation of an Object-oriented Media Composition Framework*. International Computer Music Conference, Denmark, International Computer Music Association.
- Allouche, J.P. T. J. (1995). "Finite automata and morphisms in Assisted Musical Composition." *Journal of New Music Research* 24: 97-108.
- Anderson, D. B., Jeff; (1992). *MOOD: A Concurrent C++ Based Music Language*. International Computer Music Conference, San Jose, California, International Computer Music Association.
- Anderson, T. K., Ross (1995). *Customisable Frameworks for Compositional Activity* International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Anderson, T. M. (1993). *Perceptual Parameters - their specification, scoring and control within two software composition systems* International Computer Music Conference, Waseda University, Tokyo, Japan, International Computer Music Association.
- Anderson, T. M. (1990). *E-Scape: An extendible Sonic Composition and performance environment*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Apple Computer (1994). *ResEdit*, Apple Computer.
- Assayag, G. R., Camilo (1993). *The Music Representation Project at IRCAM*. International Computer Music Conference, Waseda University, Tokyo, Japan, International Computer Music Association.
- Bach, J. S. (N.D.). *Das Wohltemperit Klavier*. Ed. Adolf Ruthardt, Leipzig. Bach, J. S. (1941). 371 Harmonised Chorales and 69 Chorale Melodies with figured bass. U.S.A, Hal Lenord Publishing Co.

- Balzer, K. S., Bernd (1990). *Counterpoint Compositions in non-tempered systems: Theory and Algorithms* International Computer Music Conference, Glasgow, International Computer Music Association.
- Bell, M. E. (1995). *A MAX Counterpoint Generator for Simulating Stylistic Traits of Stravinsky, Bartok, and Other Composers*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Bernardini, N. T, Graziano; Manfrin, Michele (1995). *The POOLman project*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Bilmes, J. (1992). *A Model for Musical Rhythm*. International Computer Music Conference, San Jose, California, International Computer Music Association.
- Bocker, H.-D. M., Andreas; Wehinher, Rainer (1990). *Beyond MIDI: Knowledge based support for Computer Aided Composition*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Bocker, H.-D. M., Andreas (1988). *What's in a Note?* International Computer Music Conference, Cologne, West Germany, International Computer Music Association.
- Bohm, C. (1996). *Automated Music, interdisciplinary aspects of a musical developmental history*. International Computer Music Conference, Hong Kong, International Computer Music Association.
- Bolognesi, T. (1983). "Automatic Composition: Experiments with Self Similar Music." *Computer Music Journal* 7(1): 25-36.
- Borgmann, A. (1984). *Technology and the Character of Contemporary Life. A Philosophical Inquiry*. Chicago, University of Chicago Press.

- Brun, H. (1985). *The premise is that there be music*. International Computer Music Conference, Vancouver, Canada, International Computer Music Association.
- Camurri, A. C., Corrado; Frixione Marcello; Innocenti, Carlo; Massucco, Claudio; Zaccaria, Renato. (1991). *A High-Level System for Music Composition*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Choi, A. (1996). *A C++ Library for Computer Music Programming on Real-Time Mach*. International Computer Music Conference, Hong Kong, International Computer Music Association.
- Coda-Music (1996). *Finale*, Coda Music.
- Cope, D. (1996). *Composers Underscoring Environment*. International Computer Music Conference, Hong Kong, International Computer Music Association.
- Cope, D. (1991). "Computer Music Intelligence in EMI" Computer Music Journal. Cope, D. (1991). *Computers and Musical Style*. Madison, Wisconsin, A-R Editions, Inc.
- Cope, D. (1990). *Pattern Matching as an Engine for the Simulation of Musical Style*. International Computer Music Conference, International Computer Music Association.
- Cope, D. (1987). *Experiments in Music Intelligence*. International Computer Music Conference, San Francisco, International Computer Music Association.
- Dannenberg, R. B. a. R., Dean (1995). *Toward Modular, Portable, Real-Time Software*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Degazio, B. (1996). *Evolution of Musical Organisms*. International Computer Music Conference, Hong Kong, International Computer Music Association.

- Desain, P. H. H. (1988). "*LOCO: a composition microworld in Logo.*" Computer Music Journal 12(3): 30-42.
- Dominique, R. M. (1994). "*Computer Music and the Post-Modern: A Case of Schizophrenia.*" Computer Music Journal 18(4): 26-34.
- Duthen, J. L., Mikael (1990). *A Compositional Environment based on PreFORM II, Patch Work and Esquisse.* International Computer Music Conference, Glasgow, International Computer Music Association.
- Eaglestone, B. L., Peter (1996). *A "Pick and Mix" approach to the Integration of Composition Tools* International Computer Music Conference, Hong Kong, International Computer Music Association.
- Eaglestone, B. M. (1996). *TEMA: An "Open II Support Environment for Music Composition".* International Computer Music Conference, Hong Kong, International Computer Music Association.
- Eaglestone, B. M. (1995). *Composition Tools Integration with a Music Database System. Database and Expert Systems Applications, DEXA 95,* London, England, Springer Verlag Lecture Notes in Computer Science 978.
- Ebcioglu, K. (1984). *An Expert System for the Shenkerian Synthesis of Chorales in the Style of J.S.Bach.* International Computer Music Conference, The Hague, Netherlands, International Computer Music Association.
- Englert, G. (1981). "*Automated composition and composed automation*" Computer Music Journal 5(4): 32-35. Englert, G. H., H (1994). "What is composition? Editor's notes." Computer Music Journal 18: 5-8.
- Fagarazzi, B. C. S. (1990). "*An Object Oriented Approach to Computer Aided Composition.*" Interface 19: 185-196.
- Ficici, S. G. (1991). *Computer Analysis of Surface Details in Tonal Music.* International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.

- Fober, D. e. a. (1996). *Recent Developments in MIDIshare*. International Computer Music Conference, Hong Kong, International Computer Music Association.
- Haig, S. H., Gershon; Harrison, Malcom. (1990). *Experience with a Harmony Learning Program*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Haig, S. H., Gershon; Harrison, Malcom C (1989). *A Shortest-path Algorithm for Musical Harmony*. International Computer Music Conference, Columbus, Ohio, USA, International Computer Music Association.
- Hailstrom, J. S., Dale (1991). *The Composer's Toolbox* International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Heidegger, M. (1977). "*The Question Concerning Technology*" and Other Essays. New York, Garland Publishing INC.
- Hitt, D. O. L., Yee (1990). *A Language for Composition*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Horry, Y. (1994). *A Graphical User Interface for MIDI Signal Generation and Sound Synthesis* International Computer Music Conference, Denmark, International Computer Music Association.
- Hunt, A. K., Ross; Orton, Richard (1991). *Musical Applications of a Cellular Automata Workstation*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Johansson, I. M., Bengt-Arne (1990). *Computer-Assisted Composition by Interactive Rhythm Transformations*. International Computer Music Conference, Glasgow, International Computer Music Association.

- Jones, D. E. (1992). *Counterpoint Assistant*. International Computer Music Conference, San Jose, California, International Computer Music Association.
- Kunze, T. T., Heinrich (1996). *SEE - A structured event editor: Visualising Compositional Data in Common Music*. International Computer Music Conference, Hong Kong, International Computer Music Association.
- Langston, P. S. (1991). "IMG/I: An Incidental Music Generator." *Computer Music Journal* 15(1): 28-39. Le Blanc, D. (1992). *Think Reference*, Symantec.
- Leonard. (1988). *Seven Centuries of Poetry in English*. Melbourne Australia, Oxford University Press.
- Leunig, M. (1993). *A Common Prayer*. Melbourne Australia, Collins Dove, Harper Collins Publishers.
- Lindemann, E. (1990). *ANIMAL - A Rapid Prototyping Environment for Computer Music Systems*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Loy, G. (1991). *Connectionism and Musiconomy*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Lunell, H. (1995). *An extendible tool for CAC*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Maguire, S. (1993). *Writing Solid Code*. Redmond, Washington, Microsoft Press.
- Mahling, A. (1991). *How to Feed Musical Gestures into Compositions*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Metrowerks (1997). *Metrowerks Code Warrior Pro*, Metrowerks Corporation.

- Mont-Reynaud, B. (1993). *SeeMusic: A tool for Music Visualisation*. International Computer Music Conference, Waseda University, Tokyo, Japan, International Computer Music Association.
- Neuhaus, T. (1995). *PPP - a Framework for Algorithmic Composition*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Ovans, R. (1990). *Music Composition as a Constraint Satisfaction Problem*. International Computer Music Conference, Glasgow, International Computer Music Association.
- Polansky, L. P. B., David Rosenboom (1990). "HMSL (*Hierarchical Music Specification Language*): A theoretical overview." *Perspectives of New Music* 28: 136-78.
- Pressing, J. (1994). *Novelty, Progress and Research Method in Computer Music Composition*. International Computer Music Conference, Denmark, International Computer Music Association.
- Pressing, J. (1993). *Visualisation and Predictive Modelling of Musical Signals using Embedding Techniques*. International Computer Music Conference, Waseda University, Tokyo, Japan, International Computer Music Association.
- Punch, B. S., Mark; Koehler, Robert (1991). *An Algorithmic Approach to Composition based on Dynamic Hierarchical Assembly*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Puckette, M. (1994). *Is there life after MIDI?* International Computer Music Conference, Denmark, International Computer Music Association.
- Riddell, A. M. (1993). *Composing the interface*, Princeton University: 334.
- Risset, J.-C. (1991). "Some Comments about Future Music Machines." *Computer Music Journal* 15(4): 32-36. Rook, V. M. (1991). *TEXTURE: A study in algorithmic composition*, University of Oklahoma: 154.

- Rossiter, D. a. H., David M. (1994). *A graphical environment for electroacoustic music composition*. International Computer Music Conference, Denmark, International Computer Music Association.
- Rowe, R. (1993). *Interactive Music Systems: Machine Listening and Composing*. Cambridge, MIT Press.
- Sagan, J. (1993). *Metro*. San Francisco, OSC.
- Schiex, T. (1992). "Possible Constrain Satisfaction Problems, or 'How to Handle Soft Constraints?'" Personal Communications CERT-ONERA.
- Sepesi, G. P., Bill (1991). *The Design of a C++ Class Library for MIDI Applications*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Sher (1988). *The New Real Book* Petaluma CA, U.S.A, Sher Music Co.
- Skrien, D. H., Jonathan (1993). *Krayola*. A Flexible, Robust and User-friendly Precompositional Sketching Environment. International Computer Music Conference, Waseda University, Tokyo, Japan, International Computer Music Association.
- Steels, L. (1986). *Learning the Craft of Musical Composition*. International Computer Music Conference, The Hague, Netherlands, International Computer Music Association.
- Sydow, D. P. (1997). *Metrowerks Code Warrior Programming*. New York, M&T Books.
- Taube, H. (1994). *Complex Musical Pattern Description in Common Music*. International Computer Music Conference, Denmark, International Computer Music Association.
- Taube, H. (1991). "Common Music: A Music Composition Language in Common Lisp and CLOS." *Computer Music Journal* 15(2): 21-32.

- Tipei, S. (1995). *For an Intelligent use of Computers in Music Composition*. International Computer Music Conference, Banff, Alberta, Canada, International Computer Music Association.
- Tipei, S. (1989). *Manifold Compositions. A (Super) Computer-Assisted Composition Experiment in Progress*. International Computer Music Conference, Columbus, Ohio, USA, International Computer Music Association.
- Tsang, C. P. A., M. (1991). *Harmonising Music as a Discipline of Constraint Logic Programming*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Various (1993). "Putting MAX in Perspective. Editor's notes (strengths and weaknesses of popular Max music programming tool)." Computer Music Journal 17: 3-11.
- Walker, J. (1997). *RICE. A Software Composers' Assistant* Australian Computer Music Association Conference, University of Auckland, New Zealand.
- Winsor, P. (1991). *PAT-PROC. An Interactive, Pattern-Process, Algorithmic Composition Program*. International Computer Music Conference, Montreal, Quebec, Canada, International Computer Music Association.
- Wyatt, D. (1997). *OMS (Open Music System)*. Palo Alto, CA, Opcode Systems Inc.
- Wyatt, D. (1997). *OMS Software Developers Kit* Palo Alto, CA, Opcode Systems Inc.
- Wyatt, D. (1990). *OMS Timing* Opcode Systems Inc. Yavelow, C. (1992). *MacWorld Music & Sound Bible*. San Mateo, IDG Books Worldwide, Inc.

- Yavelow, C. (1986). *The Impact of MIDI upon Compositional Methodology*. International Computer Music Conference, The Hague, Netherlands, International Computer Music Association.
- Zicarelli, D. (1992). *Music Technology as a Form of Parasite*. International Computer Music Conference, San Jose, California, International Computer Music Association.
- Zicarelli, D. M. P. (1995). *Max*, Opcode Systems.

APPENDIX. RICE SOFTWARE AND MANUAL

Please refer to 3₁/₄ inch floppy disk titled:

APPENDIX

RICE

and the document titled:

**RESTRICTED INTERVAL
COUNTERPOINT ENGINE**

User Manual

A Composer's Sketchpad

Version 1.0 beta

Restricted Interval Counterpoint Engine

A Composer's Sketchpad

User Manual



Version 1.0 beta

TABLE OF CONTENTS

ABOUT THIS MANUAL	4
WORKING WITH R.I.C.E. - AN OVERVIEW	5
WHAT DOES R.I.C.E. NOT DO?	6
SYSTEM REQUIREMENTS	6
CONFIGURING OMS	7
THE R.I.C.E. INTERFACE	8
• THE DISPLAY WINDOW	8
• RESTRICTED INTERVAL TABLES	10
SEARCHING FOR, AUDITIONING,	
AND ACCEPTING CHORDS	11
• THE CENTRE CONTROL	11
• EXHAUSTIVE SEARCHING	12
• AUDITIONING THE PALETTE OF 'POSSIBLE CHORDS'	14
• ACCEPTING A CHORD	15
• THE SAVE BUTTON	15
• DIRECTIONAL SEARCHES	16
ADVANCED FEATURES.	18
• THE SET MANAGER	18
• 'LOCKED' NOTES AND RESTS	20
• THE CANON MANAGER	21
• THE AUTO FUNCTION	23
EDITING FUNCTIONS	24
• EDIT MODE	24
• SELECTING PORTIONS TO EDIT	25
• EDITING COMMANDS	26

DATA INPUT	28
• MIDI-IN MODES	28
• 'INSERT LINE' MODE	29
• 'INSERT CHORD' MODE	29
• 'PERFORM' MODE	30
• IMPORTING MIDI FILES	30
THE ANALYSIS ROUTINE	32
DATA OUTPUT	33
• EXPORTING TEXT FILES	33
• EXPORTING MIDI FILES	33
• 'PERFORM' MODE	34
• THE OMS INTER-APPLICATION COMMUNICATION BUS	35
THE 'DOCUMENT SETTINGS...' DIALOG	37
SETTING THE NUMBER OF VOICES	38
LIST OF SHIFT CLICKS AND ADDITIONAL FEATURES	40
THE SCROLL BARS, WINDOW MANAGEMENT AND NOTE SIZE	42
METHODS FOR WORKING WITH R.I.C.E.	43
• UNDERSTANDING AND MANAGING THE SEARCH ENGINE	43
• MORE INFORMATION ON CONFIGURING THE INTERVAL RESTRICTION TABLES	45
• RHYTHMIC STRATEGIES FOR R.I.C.E. ASSISTED COMPOSITIONS	45
KNOWN BUGS	47
ANTICIPATED FEATURES AND IMPROVEMENTS FOR FUTURE VERSIONS	47

ABOUT THIS MANUAL.

Even though you are free to dip into this manual and read whatever you want, it is intended to be read from start to finish. It is designed this way because gaining an understanding of many of R.I.C.E.'s more complex features is dependent on concepts, methods and techniques discussed early in this manual.

This is true of most manuals, however I draw attention to this here because R.I.C.E. is a complex and specific compositional tool aimed at a relatively small audience. Design emphasis has been focused on allowing experienced users to be able to work quickly and efficiently. Because of this, and due to the nature of the tool, R.I.C.E. does not boast the classic G.U.I. ease-of-use interface for first time users. Please take the time to read this manual comprehensively to get the most out of R.I.C.E.

Most sections of this Manual assume that you are working from the stationary pad file called 'Demonstration file'. However, the section on the 'Canon Manager' uses a separate example file called 'Canon Manager Demo'.

WORKING WITH R.I.C.E. - AN OVERVIEW

R.I.C.E. is a tool which can help a composer generate material for a composition. It is designed to be integrated in the compositional process and integrated with existing commercial MIDI software.

R.I.C.E. makes the assumption that the composer is writing for a set number of 'voices' or parts, and that these 'voices' will play music notated in the traditional Western style. However, R.I.C.E. does not use traditional Western notation in its interface, nor does it hold any explicit rhythmic durational information about the music it holds. There are advantages to approach, and this manual specifically discusses rhythm in relation to R.I.C.E.

To define the sound of a composition (or sections of it), R.I.C.E. holds tables of intervals that are associated with each voice. The main task R.I.C.E. undertakes is searching for 'possible' chords which satisfy the tables of interval restrictions. These chords are visually and aurally presented to the composer to be auditioned for inclusion in a sequence of chords that will form a composition or precompositional material. The composer selects a chord, and then usually asks R.I.C.E. to search afresh from this newly selected chord.

The interval tables that define the sounds of compositions may be freely altered at any time and R.I.C.E. provides tools to manage the configuration, manipulation and storage of multiple restriction tables. These tools make the task of maintaining the integrity of the 'sound' of a composition both easy and musically intuitive.

In addition, R.I.C.E. eases the complexity of writing self similar counterpoint, and particularly facilitates composition in musical forms such as canon. Maintaining themes and harmonising precomposed material is also straightforward.

One of the most powerful features of R.I.C.E. is it's 'Performance Mode' - which lets the composer rhythmically 'play' the musical material created in R.I.C.E. straight into a sequencer or notation package.

This initial public version of R.I.C.E. is an attempt to create a powerful and integrated composers sketchpad - where the computer does the 'hack work' quickly and accurately, leaving the composer free to make musical and compositional decisions unhindered. While R.I.C.E. certainly can not cater for every compositional need, it begins bridge the gap between the human element in the compositional process and traditional 'data management' and 'graphic' programs such as sequences and notation software.

WHAT DOES R.I.C.E. NOT DO?

- Algorithmic composition.
- Create large scale form.
- Act as a traditional sequencer or notation package.
- Write functional tonal music.

SYSTEM REQUIREMENTS

The minimal requirements for R.I.C.E. are as follows.

- 680x0 processor
- 8 bit colour display (256 colours or greys)
- 2MB disk space
- 2.5 MB of free RAM
- System 7.1

Since R.I.C.E. is a computational demanding piece of software, the following setup is recommended.

- 68040 or Power PC processor
- Thousands of colours.
- 2MB disk space
- 3 MB of free RAM
- System 7.1 or greater

In addition, R.I.C.E. requires the following MIDI hardware and software.

- OMS 2.x
- MIDI interface connected to MIDI Keyboard and or Sound Box, OR,
- Quicktime 2.5, with Musical Instruments Extension.

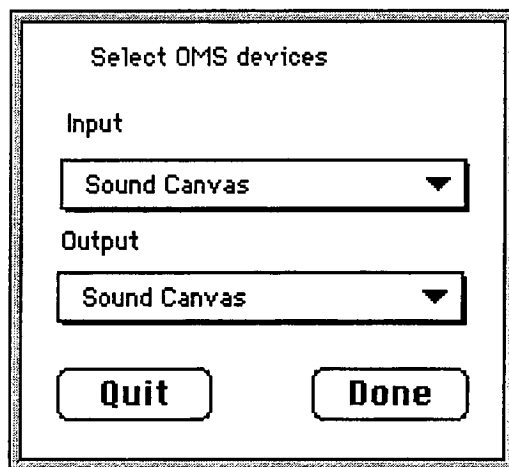
CONFIGURING OMS

OMS is Opcode's 'Open Music System' - a free MIDI Management system for desktop computers. It standardises the way MIDI data is handled on your computer, and allows MIDI applications and hardware to share data in real time. Most commercial MIDI applications support OMS.

R.I.C.E. relies on OMS to handle its MIDI input and output. To connect R.I.C.E.'s MIDI output to your keyboard, or to Apple's Quicktime Musical Instruments you need OMS. OMS is available FREE from Opcode at www.opcode.com *Note that R.I.C.E. will NOT function under any version of OMS before version 2.0.* At the time of writing the current version of OMS is 2.3.2.

R.I.C.E. will run without OMS installed, however you will not be able to input MIDI data or hear any of the music you are working on - so install OMS before getting serious! If you run R.I.C.E. without OMS you will see a dialog telling you that OMS is not installed and you will have the option to quit.

If OMS is installed R.I.C.E. presents the following dialog, asking you to specify which MIDI devices you want to use for the MIDI input and output.



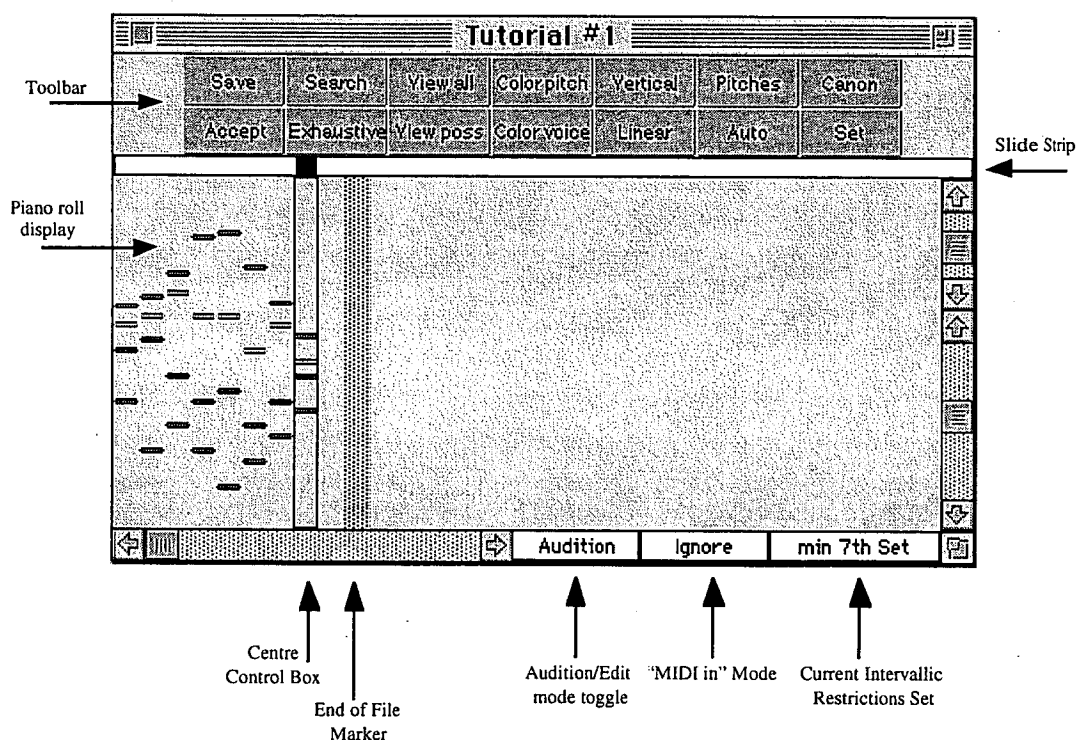
You may change R.I.C.E.'s OMS MIDI ports at any time by selecting 'OMS' from the Window menu.

If you don't have access to MIDI hardware, you can use the internal Macintosh speaker by connecting the Quicktime Musical Instrument Synth to R.I.C.E. via OMS. With the OMS Quicktime driver installed, select 'Quicktime' from the output menu in the above dialog box. Make sure that the Quicktime Musical Instrument control panel is switched 'on'! Quicktime is available free from Apple, at www.apple.com. OMS is available free from Opcode at www.opcode.com

THE R.I.C.E. INTERFACE

THE DISPLAY WINDOW


Lets jump straight into the program itself. Open the Stationary pad 'Demonstration file'. This document should be in the same folder as the R.I.C.E. application. When opened, the Display Window will look something like this.




R.I.C.E. displays music in a 'Piano Roll' format. Each small coloured rectangle represents a note. These notes can be heard by clicking on them - and clicking where there *isn't* a note will play the entire chord where you clicked. If you click and drag, you will hear all the chords that you drag over. Dragging past the edges of the window will scroll the piano roll display if required. (Please read the section 'Configuring OMS' if you can't hear anything!) Note that the closer to the top of the display window you click, the louder the MIDI output is. To play quietly, click at the bottom of the display window.

In 'Demonstration file' the notes are coloured according to the voice that they belong to. This file holds four voices, and each voice has a different colour. Voice 1 is red, 2 is green, 3 is blue and 4 is khaki. The notes in a single voice can be heard by clicking on an individual note, dragging past the right of the note and releasing the mouse button. If you want this playback to stop before it reaches the right hand side of the window, simply click the mouse button.

When the notes are coloured according to the voice they belong to, you are unable to determine what the actual pitch of any note is unless you click on it to hear it. Often you want to be able to recognise pitches visually, (as with traditional notation), and

clicking the  button in the toolbar colours each pitchclass according to the following system. Becoming familiar with this colour system will let you recognise the pitches quickly and easily.

C	White
C#/Db	Brownish Orange
D	Rich Brown
D#/Eb	Dark Blue
E	Light Blue
F	Rich Yellow
F#/Gb	Light Yellow
G	Green
G#/Ab	Dark Red
A	Red
A#/Bb	Black
B	Grey




Clicking  restores the normal colour method, where notes are coloured according to the voice they belong to.

A third colouring method is available under the 'View' menu. This is called '**Colour by voice + pitch**'. When this option is selected, the left hand two thirds of each note are coloured according to the voice it belongs to and the right hand third is coloured according to the pitch of the note.

RESTRICTED INTERVAL TABLES

R.I.C.E. holds tables of interval restrictions to define what chords it should return during searches. These interval restriction tables define the ‘sound’ of a composition or part thereof.

The restriction table windows can be called to the front at any time by using the buttons

, , or  respectively. They may also be accessed from the ‘Windows’ menu.

The following three windows describe the interval restrictions for the current set in the Demonstration file. The seven chords in the Demonstration file were generated using these tables.

Linear Intervals												
	0	1	2	3	4	5	6	7	8	9	10	11
1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Vertical Intervals												
	0	1	2	3	4	5	6	7	8	9	10	11
1-2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3-4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The ‘Linear Intervals’ table represents what intervals each voice may move up or down by. The voices are represented down the left hand side, and the intervals (in interval class notation) are represented along the top. Set out in a similar format, the ‘Vertical Intervals’ table describes what intervals may occur *between* every voice pair. The ‘Pitch Classes’ table defines what pitches are permitted in each voice.

The contents of these tables may be altered simply by clicking the checkboxes on and off. Note that clicking on the row of numbers at the top of each table will play the appropriate interval or pitch class.

R.I.C.E. provides a 'Analysis Routine' which allows these tables to be configured from existing music. The 'Analysis Routine' is detailed in the section 'Advanced Features'.

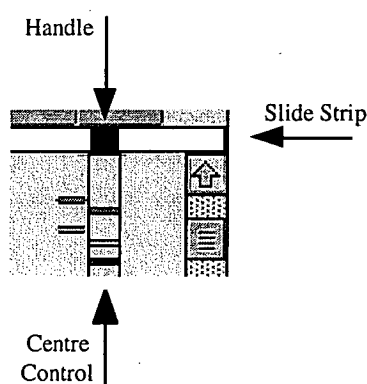
R.I.C.E. provides a 'Set Manager' for handling multiple sets of these interval restrictions tables for complex compositions. The 'Set Manager' is also described under the section 'Advanced Features'.

SEARCHING FOR, AUDITIONING, AND ACCEPTING CHORDS

Working in R.I.C.E. is normally a repeated process of searching, auditioning and selecting chords. R.I.C.E. searches return 'possible' chords that may follow an existing chord. However, before R.I.C.E. is able to search for chords that fit the current interval restrictions tables it needs a chord to search *from*.. R.I.C.E. knows which chord to search from by the position of the 'Centre Control'.

THE CENTRE CONTROL

The Centre Control is the black rectangle which outlines a single chord. It has a black 'Handle' which fits into the 'Slide Strip'. Its main function is to tell R.I.C.E. where to begin searches from.



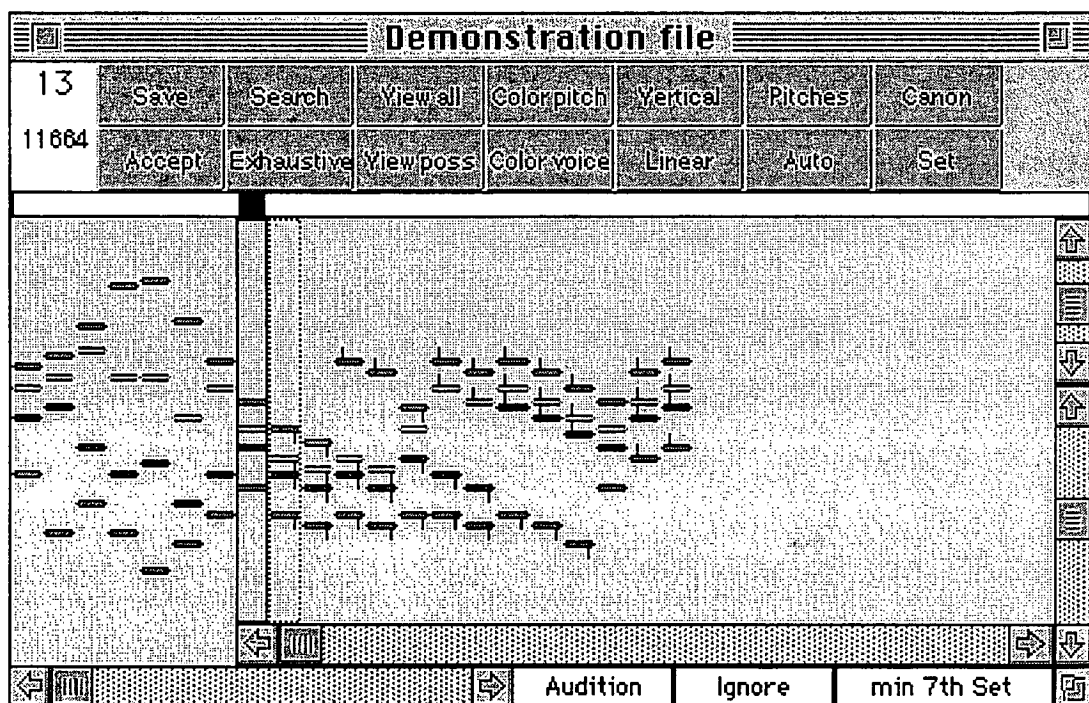
The Centre Control is repositioned by clicking on the Handle and dragging it along the Slide Strip. A dotted grey outline of the Centre Control tracks the mouse until the mouse is released. When you move the Centre Control, you are telling R.I.C.E. which chord you want to search from.

After experimenting with moving the Centre Control, return it to the last chord in the sequence, so that that the file on your computer matches the example over the page.

EXHAUSTIVE SEARCHING

Lets imagine that you want to add a new chord to the end of the sequence in 'Demonstration file'. We will ask R.I.C.E. to search exhaustively and return every chord which could follow the chord under the Centre Control. Remember the chords

returned will only be ones which fit the restrictions tables. Clicking the **Exhaustive** button will start R.I.C.E. searching for chords that can follow the chord under the Centre Control. The Slide Strip temporarily becomes a green and black progress bar. The chords are displayed to the right of the Centre Control as they are found, and the end result of this exhaustive search should look like this.



After performing a search in R.I.C.E., the display shows a palette of possible chords to be auditioned, *not* a sequence or chord progression. Even though you can still play all the chords displayed on the screen by clicking on or dragging across them, they do *not* form a sequence.

The chords to the *left* of, and under the Centre Control are indeed the original chord sequence, but the chords to the *right* of the Centre Control are a palette of possible chords - and only one chord will be chosen from the palette to be the next chord. It is essential to understand and remember this subtle difference, and a green background replaces the standard grey to remind you that R.I.C.E. is in 'View Possible chords' mode and you must interpret the chords accordingly. This 'View Possible chords' is automatically selected after any search is performed.

The stalks that have appeared on the possible chords indicate the direction that each voice moved to arrive at each chord. An additional scroll bar at the bottom of the

display has appeared to allow you to scroll though the palette of possible chords should they spill past the right hand edge of the window. The numbers to the left of the toolbar indicate that 13 of the 11664 chords that were searched conformed to the restrictions tables.

Exhaustive searching can take a long time on slower computer, particularly if there are a large number of linear intervals and a large number of voices. Pressing 'Command-Period (⌘.)' during a search gives you the option to stop the search and audition the chords that have be found so far.

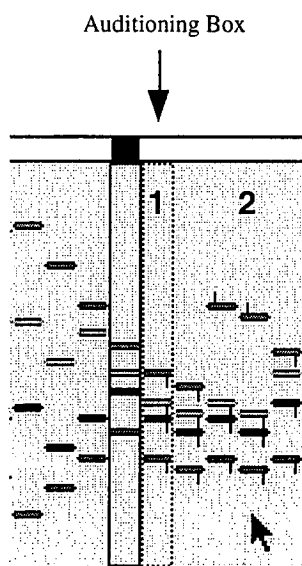
AUDITIONING THE PALETTE OF 'POSSIBLE' CHORDS

You are free to click on and hear the chords the R.I.C.E. returned from the exhaustive search. When you find a chord that you feel may be a good choice to succeed the chord under the Centre Control, you need to put it into the 'Auditioning Box'. The Auditioning Box is the dotted grey box that has appeared to the left of the Centre Control.

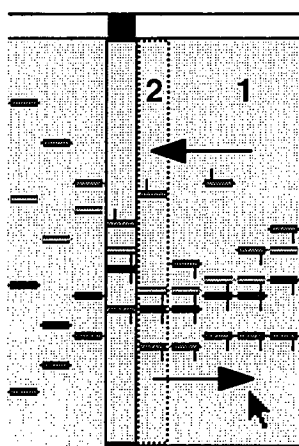
To do this, Shift-Click a chord in the palette of possible chords, (to the right of the Centre Control). You will see that the chord you clicked has been swapped with the chord in the Auditioning Box. This allows you to drag across the chords from the left of the Centre control to the chord in the Auditioning Box, so that you can hear the chord you selected from the palette in the context of the existing chord progression.

If you are unhappy with your selection, simply repeat the Shift-Click procedure on another chord from the palette of possible chords, and this new chord will be swapped with the old chord in the Auditioning Box. Then drag across the chords from the left of the Centre Control to hear this new chord in context.

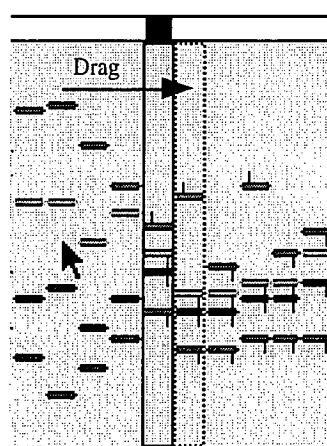
Before the Shift-Click




After the Shift-Click,
the chords are swapped

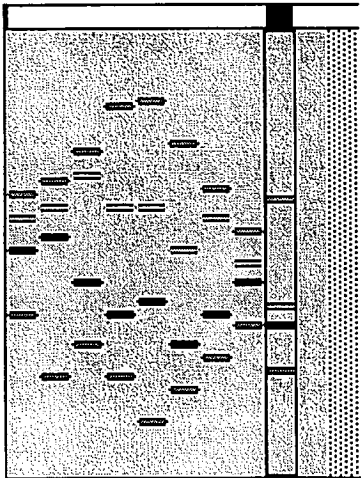


Drag across the sequence
to the Auditioning Box






ACCEPTING A CHORD

When you are sure that the chord in the Auditioning Box is the one that you want, click the  button in the toolbar. You will see that R.I.C.E. includes the chord from the Auditioning Box in the sequence and repositions the Centre Control over the newly accepted chord ready for a new search.





Notice that the background has become grey again, indicating that the display is showing the sequence instead of the palette of possible chords. Although R.I.C.E. changes between these two modes automatically, sometimes when you are placing new chords over old ones, you want to see the entire sequence. There are two buttons,

 and  which change between showing all the chords in the sequence, and showing the palette of possible chords. Note also that scrolling the sequence while in 'View poss' mode changes the position of the Centre Control, clears the palette of possible chords, and returns R.I.C.E. to 'View all' mode.

Refer to the section detailing the  feature for a rudimentary method of forcing R.I.C.E. to generate material unattended.

THE SAVE BUTTON

The  button is conveniently located above the  button so that it is easy to save your work periodically after each newly accepted chord.

Normal 'clickable' R.I.C.E. files have the  icon.

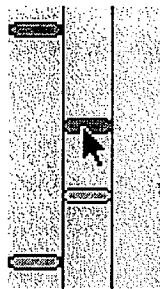
DIRECTIONAL SEARCHES

In the last section R.I.C.E. performed an 'Exhaustive' search. It returned *every* chord to the palette of possible chords. Sometimes you may not want to see every possible chord. For example there may be too many (sometimes even hundreds) to sort through, audition, and finally select just one. Alternately you may only be interested in the possible chords that are formed when the individual voices form a specific contour; that is they move in specific directions.

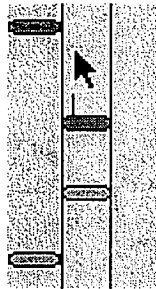
'Directional' searches allow you to specify to R.I.C.E. to return only a subset of all the possible chords by indicating which directions you want the voices to move in. You tell R.I.C.E. which direction you want each voice to move in by putting directional 'stalks' on the chord under the Centre Control.

To put a stalk on a note under the Centre Control, click on a note and then drag above or below it, depending on whether you want the voice to move up or down.

Click on the
note...



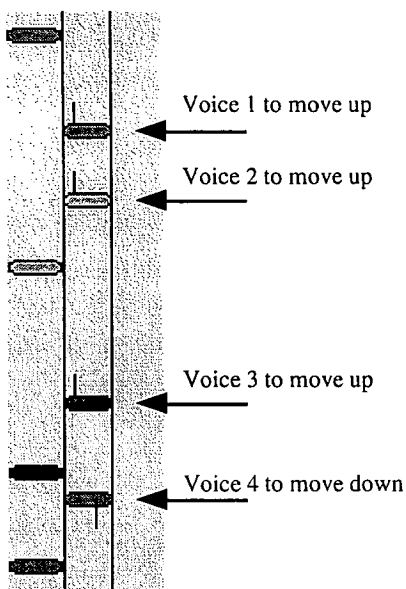
... drag up (or down)
and release the mouse button.




If you simply click on a note in the Centre Control no stalk will be added, and any stalk that was present will be removed. The absence of a stalk tells R.I.C.E. that that voice is to remain on the same note in any chords returned in the palette of possible chords.

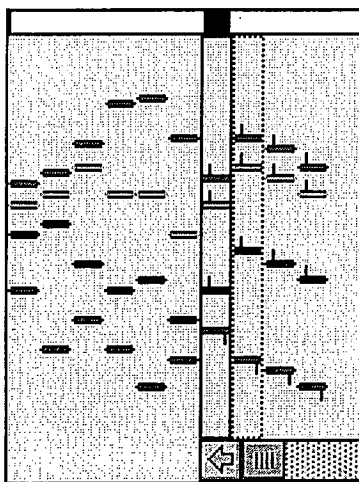
If you have R.I.C.E. connected to a MIDI output device, you will hear aural confirmation of the direction that you have indicated. If you drag up on a note, you will hear the same note an octave above when you release the mouse. Similarly, you hear an octave below for dragging down. If you just click on the note, the same note is repeated when the mouse is released.

To perform a directional search, reopen the Stationary pad 'Demonstration file' and add stalks to the chord under the centre control so they look like the example below. (You may try any directions you like, but these directions should definitely yield results in this case!)



(Note that you cannot put stalks on chords outside the Centre Control. Recall that R.I.C.E. puts directional stalks on the chords which form the palette of possible chords to indicate what direction each voice moved to form that chord).


Now, click the  button, (not 'Exhaustive') and the following should appear very quickly.

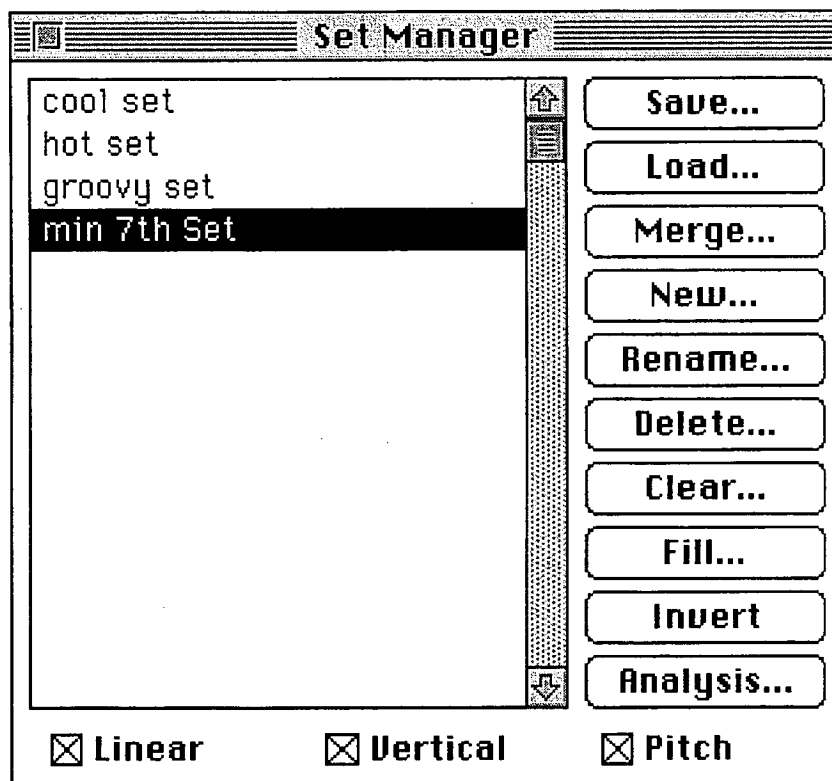


Directional searches return a palette of possible chords which may be auditioned in exactly the same way as after the Exhaustive search. Remember that when using directional searches, you are often severely limiting the number of possible chords that will be returned, and frequently there may be none that fit!

ADVANCED FEATURES.

THE SET MANAGER

The Set Manager allows you to maintain and edit multiple sets of interval restrictions tables. Click the  button in the toolbar, (or select 'Set' from the 'Window' menu) and the following window will appear.



The 'Demonstration file' has four interval restriction tables preloaded, each with names referring to different Jazz-style harmonies. The available interval restriction tables appear in the list on the left hand side of the Set Manager Window. Only one set of interval restriction tables may be selected at a time, and this is done by clicking on the one you want. You will notice that the restriction table windows update immediately and automatically.

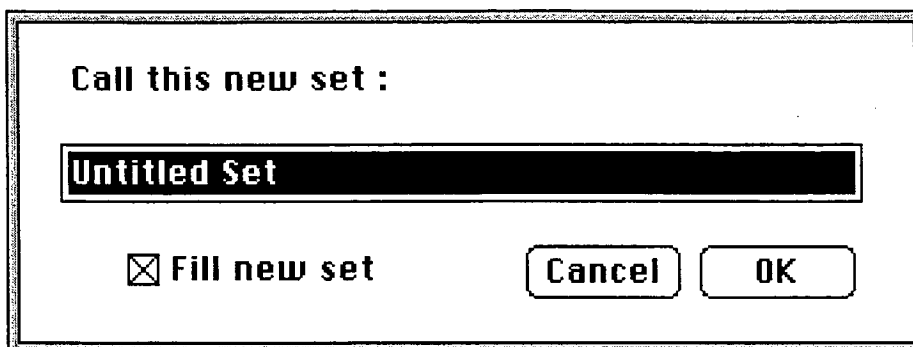
The buttons on the right side of the window perform actions on sets of interval restriction tables or on groups of sets. Entire sets may be loaded, saved and merged independently of R.I.C.E. music data, allowing interval restriction tables that have been created in one file to be loaded into another.



R.I.C.E. 'Set' files have the icon , and are non clickable.

The 'New' button in the Set Manager Window presents this dialog before creating a

new interval restriction table.



Uncheck the 'Fill new set' checkbox if you want the new tables to be 'blank'.

The 'Rename' and 'Delete' buttons act on the currently selected interval restriction table in the list on the left hand side of the window. The 'Clear', 'Fill' and 'Invert' buttons act on the checkboxes of the currently selected interval restriction set. *Note that none of these actions are undoable!* The 'Analysis' button provides access to the analysis routine described later.

Three checkboxes appear at the bottom of the Set Window.

☒ **Linear** ☒ **Vertical** ☒ **Pitch**

They represent which specific interval restriction tables the buttons 'Clear', 'Fill', 'Invert' and 'Analysis' should apply to. For example, if you want to invert the Vertical intervals but not the Linear intervals or the Pitches, the checkboxes should look like this.

☐ **Linear** ☒ **Vertical** ☐ **Pitch**

When working in R.I.C.E., you will often find that you want to change sets frequently. There is a popup menu at the bottom right hand corner of the Display Window, which shows the name of the current interval restrictions table.



Clicking on this menu allows you to quickly select any other available set of restrictions, or create a new set without opening the Set Manager Window.



Now that you have know how to change the sets, go back to the document 'Demonstration file' and select a different set to the 'min 7th Set', and ask R.I.C.E. to perform an exhaustive search. You will hear the difference in sound in the possible

chords - they are no longer minor sevenths!

'LOCKED' NOTES AND RESTS

R.I.C.E. allows you to 'lock' notes. This is useful for a numbers of reasons. Firstly, it is a way of ensuring that notes and chords that you have generated are not easily altered. The simple way to lock notes is to hold down the 'Control' key and click on a note. The note becomes a shaded grey of its original colour, denoting that it is locked. Control-Clicking the note again will unlock it.

The most important function of locked notes involves the Centre Control and the Searching. When a chord immediately to the *right* of the Centre Control contains a locked note, the Search routines are forced to include that note in the possible chords.

This means that locked notes are particularly useful when you are harmonising precomposed musical material. This is discussed in the section 'Methods for working in R.I.C.E.' Locked notes are also important in the operation of the 'Canon Manager' (discussed below).

R.I.C.E. also allows rests in the music data. This simply means that a note is missing from a chord. Rests are simply absent notes. Option-Clicking a note will delete that note. *Deleting notes by this method is NOT UNDOABLE!*

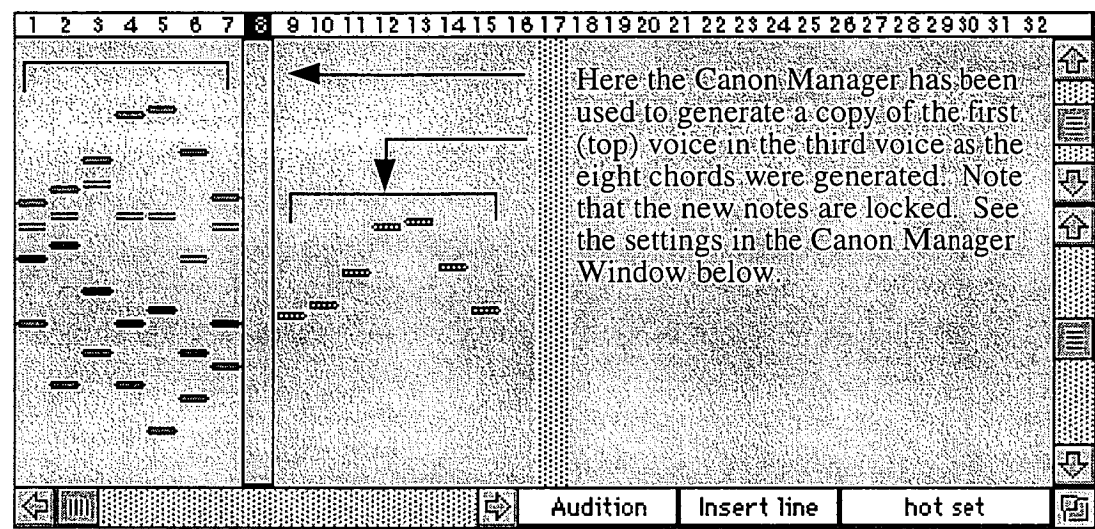
When the chord under the Centre Control is incomplete, (that is, it contains a rest) the search engine will look back to find that last note that occurred in that voice. It is not recommended that the first chord in the sequence contain a rest.

THE CANON MANAGER


The Canon Manager is a tool the help you write canon and self similar counterpoint. It can also be used to preserve thematic unity in your composition.

The ‘Canon Manager’ works by generating locked notes *ahead* of the Centre Control. When the canon manager is configured to generate canonic material, it inserts a new note ahead of the Centre Control every time a new chord is accepted. When the Centre Control catches up with the locked notes generated by the Canon Manager, they are incorporated into the search engine results, and into the composition.

Open the Stationary pad file called ‘Canon Manager Demo’.



In this example, the chords to the right of the Centre Control were generated by the canon manager when the 7 chords to the left of the centre control were accepted. These notes are locked so that the Search Engine will be forced to include them in the palette of possible chords.

Click the  button. The following window will appear.

Canon Manager							
on	Source	Dest.	Rhythmic	Trans-	Aug-	Inverton	Focus
off	Voice	Voice	Offset	pose	ment	off	pitch
<input checked="" type="checkbox"/>	1	3	8	-18	1	<input type="checkbox"/>	C
<input type="checkbox"/>	4	2	8	20	1	<input type="checkbox"/>	C
<input type="checkbox"/>	1	1	1	0	1	<input type="checkbox"/>	C
<input type="checkbox"/>	1	1	1	0	1	<input type="checkbox"/>	C

The Canon Manager Window allows configuration of as many canon as there are voices. Canon are turned on and off with the checkboxes at the left of each row of

controls.

The material for a canon must come from a 'Source Voice', and go to a 'Destination Voice'. There may be any combination of voices you choose. In the first canon configuration here the canon is sourced in voice One and is destined for voice Three.

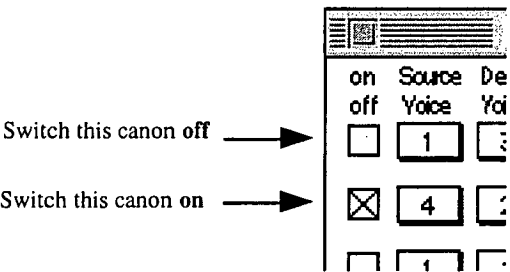
The 'Rhythmic Offset' menu tells R.I.C.E. how far ahead to place the notes in the canon. The number refers to how many rhythmic events, (chords), ahead of the Centre Control this canon should appear. When the Canon Manager Window is open, R.I.C.E. displays the number of each chord in the Slide Strip. This is to help you calculate the Rhythmic Offset.

The 'Transposition' menu tells R.I.C.E. if and how the canon should be transposed. The numbers represent semitones away from the original voice. Dragging down the menu will transpose down and vice versa. Selecting '0' will cause the canon to be untransposed. In the example above, the first canon is configured to transpose down an octave and a tritone.

The 'Augment' feature is to be used to make Rhythmically augmented canon. Selecting a value larger than one will magnify the rhythmic duration of the canon. This feature is not available in R.I.C.E. version 1.0. Only '1' can be selected.

The last option for configuring a canon is to 'Invert' the notes of the canon. You switch this feature on or off by using the checkbox at the right hand side of each row of controls. The 'Focus pitch' tells R.I.C.E. which note to invert the notes of the canon about. The Invert feature inverts notes chromatically about this focus. For example, if Middle C is the focus, and an 'A' above Middle C comes through the Canon Manager, the 'Eb' below middle C will be output. 'A' is a minor sixth, (Interval Class 9) above Middle C, and 'Eb' is a minor sixth below Middle C. Note that inversions take place *after* any transposition has been performed.


To experiment with the canon manager yourself, switch the first canon settings off, and switch the second canon settings on.





Note that the selected set of interval restrictions has been changed from 'min 7th Set' to 'hot set'. This means that you will be harmonising the melody in voice one from the first 7 chords in voice three with a different set of restrictions, and a different sound.

Repeat the Search -> Audition -> Accept loop with the new Canon Manager configuration, to get an idea of how the Canon Manager works in practice. You will observe that the palette of possible chords is much smaller, because it only contains chords that have the locked notes in them. You are able to see the results of the current Canon Manager settings after accepting a chord by switching between 'View all' and 'View poss' modes using the buttons in the toolbar if required

THE AUTO FUNCTION

The  button forces R.I.C.E. to perform the Search -> Audition -> Accept loop on its own. First, R.I.C.E. searches exhaustively and then selects a chord from the palette of possible chords at random. This chord is accepted, and after a delay of a second, the procedure is repeated.

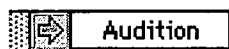
The Auto mode may be stopped by clicking on the  button during the seconds pause between each search, or by pressing 'Command-Period ()' during a search.

The Auto mode is designed to allow you generate data quickly, and hear the 'sound' and check the feasibility of a certain table of restrictions. It is *not* intended to be a successful method of algorithmic composition, since the selected chords are purely random.

EDITING FUNCTIONS

EDIT MODE

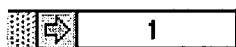
R.I.C.E. provides a suite of editing facilities for manipulating the musical data. Most editing of musical data takes place in the 'Edit' mode. Normally R.I.C.E. is in 'Audition' mode. The Audition mode is denoted by a grey or green background, depending on whether you are viewing all the chords, or just the palette of possible chords. A pop up menu at the bottom of the display window also indicates the Audition mode.



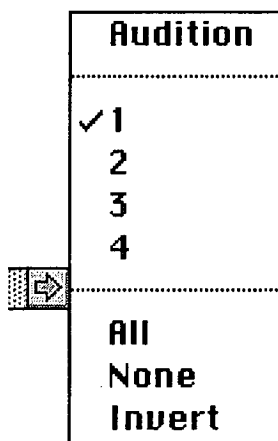
To change from 'Audition' mode to 'Edit' mode, click on this pop up menu and select 'Edit'.



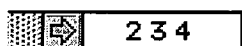
The display of the menu will should become '1'. Numbers in this menu display indicate the voice that will be edited during 'voice specific' operations. 'Voice specific' operations do not act globally on all voices.



Notice that the display window's background has become red, to indicate that R.I.C.E. is now in 'Edit' mode. Click on the same menu again, and you will observe the the contents has changed.



Selecting 'Audition' returns R.I.C.E. to the normal auditioning mode. Selecting one of the voice numbers toggles the check next to the number, indicating whether that voice will be included in voice specific editing operations. Choosing 'All', 'None' or 'Invert' acts on the selection of the voice numbers. Any changes in voice number are reflected in the pop up menu's display when the menu is released. For example, selecting 'Invert' in the above menu will yield this result.



Note that not all editing functions are voice specific. Some operations act on entire chords regardless of what voices are selecting under this menu.

SELECTING PORTIONS TO EDIT

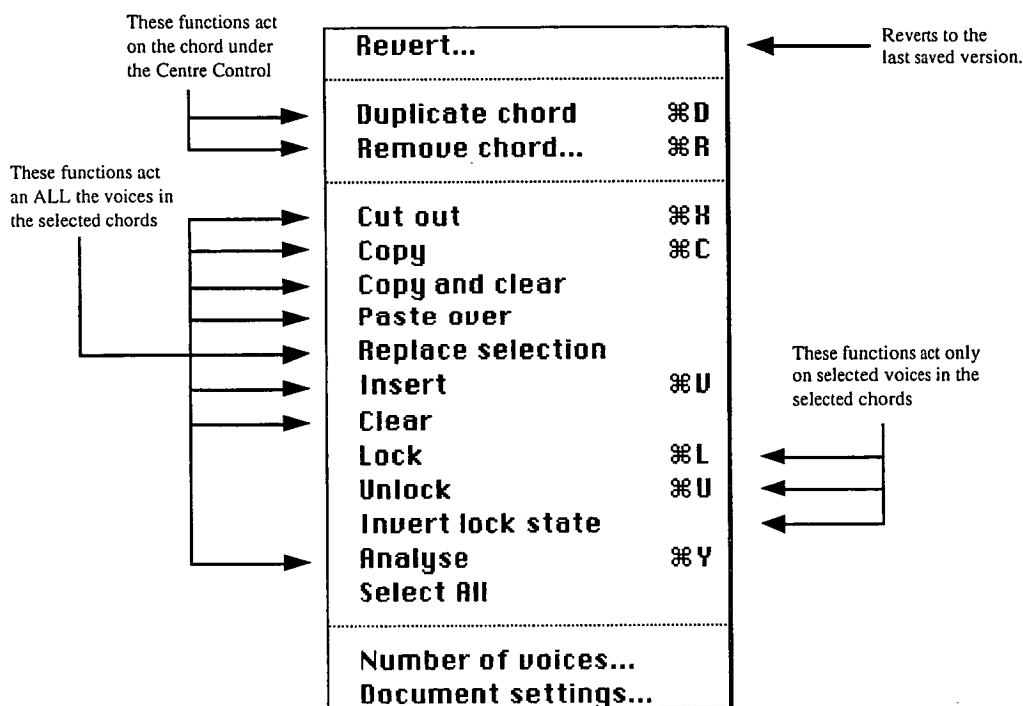
Selected chords in Edit mode are on a blue background. You should see that the first chord in the sequence is coloured with a blue background. Clicking and dragging to the right across chords in 'Edit' mode will select chords. You will also see a green boundary at the start of the selection and a red boundary at the end.

You can drag past the end of the display window, and the display will scroll, selecting chords as they appear. You can alter the right hand boundary of the selection by Shift-Clicking anywhere to the right of the green starting boundary.

A '**Select all**' function is available under the 'Edit' menu. Selecting all automatically puts R.I.C.E. into the 'Edit' mode.

EDITING COMMANDS

The Edit Menu provides a selection of editing functions. The table below indicates which editing functions are 'editable voice specific'.



Revert... provides you with the option of returning to the last saved version of your file. This function replaces the normal **Undo** function.

Duplicate chord (⌘D) takes a copy of the chord under the Centre Control and inserts immediately to the right of the Centre Control. No data is lost, and all the chords to the right of the Centre Control 'Shuffle out' to accommodate the new chord.

Remove chord... (⌘R) deletes the chord under the Centre Control. All the Chords to the right of the Centre Control 'Shuffle in' to take up the extra space. The Centre Control does not normally move with this operation, and the chord that was immediately to the right of the Centre Control is positioned under the Centre Control. *Data is lost as a result of this function and it's actions are not undoable.*

Cut out... (⌘H) cuts the selected chords out of the sequence and stores them on the clipboard. The selected chords are actually removed from the sequence, and the chords to the left and right of the selected portion abut after this operation. Locked notes *are* cut out. The size of the selection is unchanged after this operation. *Note that the linear integrity of the sequence may be interrupted. Data is lost as a result of this function and it's actions are not undoable.*

Copy (⌘C) copies the selected chords to the clipboard.

Copy and clear (⌘C) copies the selected chords to the clipboard, and then clears the selection. Locked notes are copied but not cleared. *Data is lost as a result of this function and it's actions are not undoable.*

Paste over places the contents of the clipboard over the currently selected chords starting from the left hand boundary of the selected chords. **Paste over** continues pasting the contents of the clipboard until the entire contents is pasted, regardless of the length of the selection size. If the clipboard contents is *shorter* than the chord selection size, the **Paste over** function resizes the selection to the size of the clipboards contents that has just been pasted in. Locked notes are overwritten. *Data may be written over as a result of this function and it's actions are not undoable.*

Replace selection pastes the contents of the clipboard over the selected chords. If the contents of the clipboard is longer than the selected chords, **Replace selection** truncates the pasted version of the clipboard to the size of the selected portion. If the size of the clipboard is smaller and the selected portion, **Replace selection** pastes the contents of the clipboard, leaving the rest of the selection untouched. **Replace selection** does not resize the selection boundaries. Locked notes are overwritten. *Data may be written over as a result of this function and it's actions are not undoable.*

Insert (⌘U) is the standard 'paste' function. This function inserts the contents of the clipboard at the green left-hand boundary of the selected portion. *Note that the linear integrity of the sequence may be interrupted. Although data is not lost as a result of this function, it's actions are not undoable.*

Clear... clears the contents of the selection portion. Locked notes are not cleared. *Note that the linear integrity of the sequence may be interrupted. Data is lost as a result of this function and it's actions are not undoable.*

Lock (⌘L) locks notes in the selection boundaries. **Lock** only locks notes in voices checked under the 'Edit' popup menu at the bottom of the display window.

Unlock (⌘L) unlocks notes in the selection boundaries. **Unlock** only unlocks notes in voices checked under the 'Edit' popup menu at the bottom of the display window.

Invert lock state toggles the locked/unlocked state of the notes in the currently selected voices in between the selection boundaries. All notes in the checked voices that were locked are unlocked, and all notes that were unlocked are locked.

Analyse configures the current interval restrictions tables according to the interval content of the selected portion of the sequence. It acts on all voices. *This function alters the contents of the current interval restrictions tables and is not undoable.*

Select all selects the entire sequence in R.I.C.E. by placing the selection boundaries at the start and the end of the sequence.

Number of voices... presents a dialog allowing you to alter the number of voices that R.I.C.E. is working in.

Document settings... Allows you to alter the behaviour of some of R.I.C.E.'s functions.

The **Analyse**, **Number of voices...**, and **Document settings...** functions are discussed in more detail below.

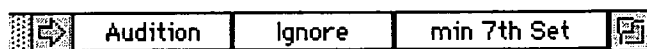
DATA INPUT

It is not enough that R.I.C.E. simply generates palettes of musical material. Often you need to get musical data into R.I.C.E. so that you can modify it, work with it or analyse it. This is usually at the start working on a section of a composition, but you may need data entry at any stage of your work with R.I.C.E.

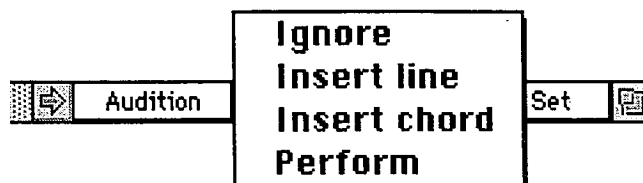
MIDI-IN MODES

R.I.C.E. provides several modes for accepting 'Live' MIDI data from an OMS port. This data will typically come from a MIDI controller such as a keyboard, but may come from another application through OMS's Inter-Application-Communication Bus, (the IAC Bus).

These modes are set using the middle pop-up menu at the bottom of the display window.

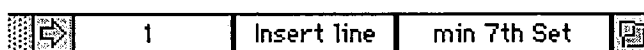


The default mode is 'Ignore', and R.I.C.E. discards any incoming MIDI data in this mode. Clicking the Midi-in mode popup menu looks like this.



‘INSERT LINE’ MODE

Selecting either **Insert line** or **Insert chord** automatically puts R.I.C.E. into ‘Edit’ Mode. The voices which are currently being edited are shown in the Audition/Edit popup menu.

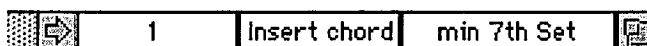


When in **Insert line** mode, R.I.C.E. places any MIDI notes received through the OMS port into the Sequence. R.I.C.E. begins inserting chords from the green left hand boundary of the selection, and advances the selection automatically so that a stream of notes can be captured.

The **Insert line** mode disregards channel information in the MIDI notes. If you want to insert notes in another voice, alter the editable voice settings in the Audition/Edit popup menu.

Note that you cannot insert notes in more than one voice at a time. If R.I.C.E. detects a multiple voice selection when in **Insert line** mode, it will inform you that it is forcing the number of editable voices to a single voice only. To change from one editable voice to another singular editable voice, hold down the ‘Shift’ key when selecting a new voice. The old editable voice(s) will be unchecked and only the newly selected voice will be checked.

‘INSERT CHORD’ MODE



The **Insert chord** mode allows you to play chords directly into R.I.C.E. from your keyboard or MIDI controller. Like **Insert line**, **Insert chord** inserts the MIDI data it receives at the beginning of the edit selection. Remember R.I.C.E. automatically selects ‘Edit’ mode when in **Insert line** mode. (Yes, R.I.C.E. is in multiple modes at once!).

Chords need to be received in R.I.C.E. in a strict format. For each chord, the first note R.I.C.E. receives is treated as the note belonging to voice one, then voice two, and three etc. This means you have to play your chords from ‘top down’, each note just the once. If you become confused, or make a mistake with a certain chord you are entering,

simply reselect **Insert chord** from the MIDI-In popup menu and R.I.C.E. will be ready to receive the top note in the chord.

Unlike **Insert line** mode, R.I.C.E. does *not* automatically advance the edit selection after every chord. This would be too confusing, since errors in this type of entry are common. You need to advance the selection yourself, either by clicking the next chord position with the mouse, or tapping the space bar. Using the space bar to alter the position of the edit selection boundaries only works in **Insert chord** mode.

When in **Insert line** mode, R.I.C.E. ignores the channel information of the incoming MIDI notes.

‘PERFORM’ MODE

The **Perform** mode treats any incoming MIDI note data as a trigger. When a ‘Note On’ MIDI message is received, R.I.C.E. plays the next chord in the sequence until a ‘Note Off’ message is received. As a result, this mode lets you ‘Perform’ your music directly from R.I.C.E. itself.

The full implications of the **Perform** mode in R.I.C.E. are so useful that this MIDI input mode is discussed in detail in its under ‘Data Output’.

IMPORTING MIDI FILES

In addition to accepting ‘Live’ MIDI data, R.I.C.E. is able to read industry standard MIDI files. However, MIDI files that are intended for R.I.C.E. to read must be specially prepared. This is because R.I.C.E. stores and manipulates musical data as separate voices without any explicit durational information.

MIDI files that are to be read by R.I.C.E. must be of format type 1, and must contain musical data made of monophonic tracks. R.I.C.E. treats each track in a MIDI file as a separate voice, as voices in R.I.C.E. are monophonic. R.I.C.E.’s interpretation of MIDI files that contain polyphonic data in the one track will be unreliable. R.I.C.E. disregards channel information when importing MIDI files.

After selecting ‘**Import MIDI file...**’ from the File menu, you will be informed that you will be asked to close any open document. If required, you will have the opportunity to save your work. R.I.C.E. will present you with the following dialog prior to asking you which MIDI file you want to load.

While importing this MIDI file...

☒ Lock note attacks

☐ Lock unstruck notes

☐ Merge with current data

Cancel

Proceed

R.I.C.E. gives you the option to lock notes that are read in from a MIDI file. This can be useful if you are reading in music that you want to harmonise. You have the option to **Lock new note attacks**, which is any change of note in each voice.

You will have observed that R.I.C.E. stores no tied notes. Notes are restruck when they are repeated, - there is only one unit of duration in R.I.C.E. Checking **Lock unstruck notes** will tell R.I.C.E. to lock any notes that are actually unstruck in the MIDI file. This happens when one voice moves against another, creating a new chord, but not a newly struck chord.

Merge with Current data is not defined in version 1.0

THE ANALYSIS ROUTINE

The key to the ability of R.I.C.E. to generate chords with a certain 'sound' is in the configuration of the interval restrictions tables. You are free to configure these tables as you wish, for example, according to some compositional scheme. However, if you have already attempted this, you will have discovered that configuring these tables manually in a way that allows searches to yield chords is *not* a trivial task.

To solve this problem intuitively, R.I.C.E. provides a powerful analysis routine for configuring the interval restrictions tables automatically. To invoke the Analysis routine, select some chords in 'Edit' mode by dragging to the right across some chords. (These will typically be chords that you have entered in one of the 'MIDI-in' modes). Then simply select the '**Analysis**' menu item under the Edit menu in the Menu bar. The three restrictions tables will be configured according to the intervals in the music selected.

Recall that in the Set Manager, there are three checkboxes at the bottom of the 'Set' window.

☒ **Linear** ☒ **Vertical** ☒ **Pitch**

The Analysis routine uses the states of these checkboxes to determine what kinds of analysis need to be undertaken. For example, unchecking the 'Pitch' checkbox will leave the pitch class restrictions table untouched after invoking the analysis routine.

The advantages of using the Analysis routine are significant. Because you are analysing *existing music* you can be fairly sure that you will have a set of restrictions that will allow you to generate music. It is not guaranteed, but it is more assured than creating a restrictions scheme 'from scratch' which not only requires a large amount of trial and error, you can not intuitively 'hear' the intervals you are setting up.

If you choose to arbitrarily modify the data in your sequence of chords, and as a result you know that it no longer conforms to the current restrictions table, select the appropriate portion of the sequence, and analyse it. The intervals restriction tables will be modified accordingly to accommodate your arbitrary change.

Compound intervals are reduced to the simple intervals.

DATA OUTPUT

R.I.C.E. provides three ways of exporting musical data.

EXPORTING TEXT FILES

The **Export Text File...** menu item exports the musical data held in R.I.C.E. to a text file. This file may be opened by any word processor. The format of the text output is as follows.

V 1	V 2	V 3	V 4	V 5
75 Eb	70 Bb	68 Ab	60 C	42 F#
69 A	66 F#	64 E	61 C#	43 G
68 Ab	63 Eb	59 B	58 Bb	41 F
68 Ab	65 F	60 C	58 Bb	42 F#
75 Eb	72 C	70 Bb	65 F	51 Eb
68 Ab	65 F	61 C#	60 C	42 F#
67 G	62 D	60 C	57 A	41 F
67 G	64 E	59 B	57 A	41 F
70 Bb	67 G	63 Eb	60 C	32 Ab
69 A	64 E	62 D	59 B	31 G

Each chord is represented horizontally, and the voices are tabulated in columns. The chords are collected into sets of five to make it easy on the eye when reading. Each note appears with it's letter name and the MIDI note number and you can tell the direction of each voices movement from the MIDI note number. A rest in a voice is represented by -1.

Text Export can be useful if you are using the material generated in R.I.C.E. for precompositional purposes. R.I.C.E. cannot import these text files.

EXPORTING MIDI FILES

The **Export MIDI file...** menu item exports the musical data held in R.I.C.E. to a Standard MIDI File. This file may be opened by any MIDI capable software. The MIDI files generated are of Type 1, and each voice's data is written in a separate track, so R.I.C.E. can read back it's own MIDI files. The chords are written out in block chords, to a straight crotchet (quarter note) rhythm, in a 4/4 time signature.

Exporting R.I.C.E.'s musical data to a MIDI file is a useful way of getting your musical material into another MIDI capable program where you may manipulate is as raw data. It is also a useful way of saving your work across changes to the native

R.I.C.E. file format, which may occur between versions.

‘PERFORM’ MODE

The ‘Perform’ mode that R.I.C.E. offers is the most powerful and flexible method of data output. It allows you to use your MIDI controller to ‘play’ your chord sequence out to MIDI hardware, or to other OMS MIDI capable software. The implications of this capability are significant.

For example, you may ‘play’ your R.I.C.E. chord sequence out to a sequencer, where you could then include that performance in your sequencer composition. Alternatively you may direct the output to a notation package, synchronising your performance to it’s metronome, thereby allowing transcription of your performance. This ‘Perform’ mode also allows R.I.C.E. to become a performance instrument - and MIDI data may be routed via manipulation software such as Max to enhance the performance. The ‘Perform’ mode allows R.I.C.E. to be integrated in the compositional/performance cycle, greatly facilitates score preparation, and fleshes out an intuitive compositional process.

‘Perform’ mode is engaged by selecting the Perform option from the middle popup menu at the bottom of the display window.

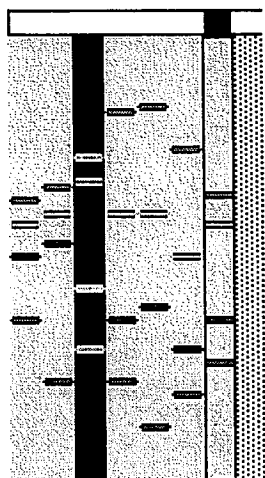


When in perform mode, R.I.C.E. is waiting for MIDI note data via its OMS input port. When it receives a ‘Note-on’ message, it will play the first chord in the sequence. The subsequent chords are played whenever a new ‘Note-on’ message is received.

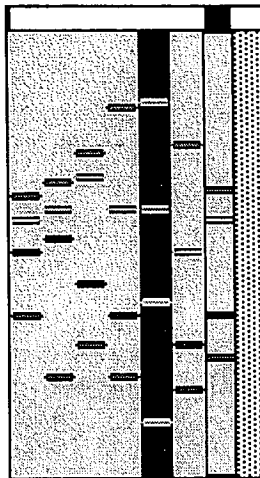
So, the ‘Perform’ mode is a simple tool to master. Simply play single notes of your keyboard or MIDI controller, and R.I.C.E. will play the chord sequence according to the rhythm that you perform. R.I.C.E. will continue to play a chord as long as a note from your MIDI controller is played. If you play a new note before the old one is released, the two chords from R.I.C.E. sound together. Releasing any note on your controller will force R.I.C.E. to switch off all the chords that are playing.

R.I.C.E. provides visual feedback during the 'Perform' mode. Whenever a chord is played, R.I.C.E. highlights that chord in the display window.

Three chords from the start.



Two chords further on.



Note that you only need to play *one note* to trigger R.I.C.E. to play a whole chord. There is not much practical application in playing chords on your keyboard in 'Perform' mode.

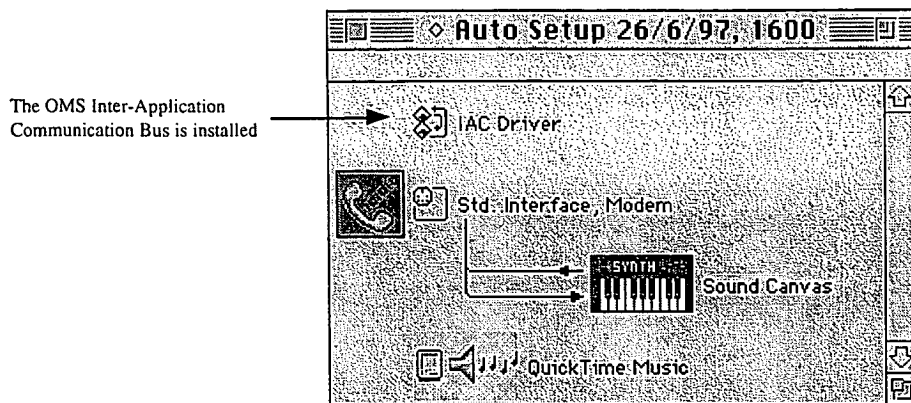
R.I.C.E. will track your dynamics. For example if you play loud, the chord output by R.I.C.E. will be the same volume.

It is probable that for most users, the MIDI keyboard used to trigger R.I.C.E. will also be the sound source for R.I.C.E.'s output. Typically this means that the note that you play will be heard at the same time as the chord that R.I.C.E. outputs. If you don't want this to happen, check to see if your MIDI keyboard has a 'Local on/off' switch. If so, switch Local 'off'. Your MIDI keyboard will behave exactly as before, except that it won't play sounds when you play the keyboard, but it *will* respond to notes coming in it's MIDI in port. This way you can play your keyboard, and hear only the chords that R.I.C.E. outputs.

THE OMS INTER-APPLICATION COMMUNICATION BUS

To connect R.I.C.E.'s data output in 'Perform' mode to another application, you need to use OMS's Inter-Application Communication Bus, (IAC Bus). A 'Bus' is computer jargon for a data pathway. You use the IAC Bus to create a data path between applications.

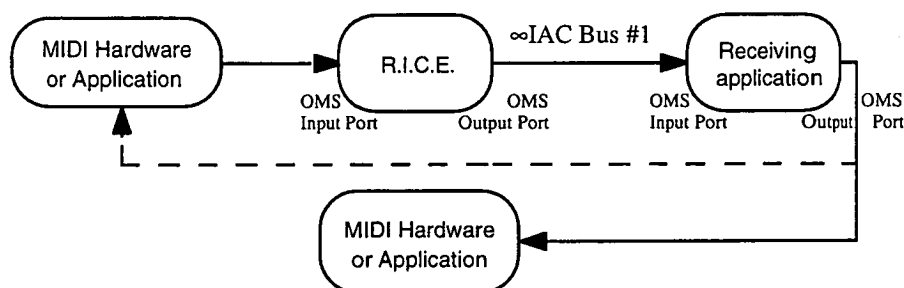
The IAC bus is automatically installed with OMS. To make sure that it is installed on your computer, open the application 'OMS Setup' (which will be installed on your hard drive), and assuming you have already configured OMS, you should see a window similar to the following example.



If you see the Icon for the IAC Driver, everything is fine. If not, reinstall OMS.

The IAC Driver is simple to configure. Think of it as a cable or 'patching lead' between two applications, and you need to plug an end into each application.

In R.I.C.E., you need to set the OMS *output* device to the '∞IAC Bus #1'. Do this in the 'Select OMS Devices' dialog which appears at startup, or select '**OMS**' from the Window menu. Then, in the application that will receive R.I.C.E.'s output, set the *input* device to '∞IAC Bus #1'. Put the receiving application in the background, bring R.I.C.E. to the foreground and 'Perform'. The data that R.I.C.E. outputs will now be sent to the receiving application. *The configuration of the receiving application's handling of the incoming data is your responsibility.* R.I.C.E. has no control over the settings of any other application.



You may need to route the output from your receiving application to a synthesiser to hear your performance. This typically involves selecting a 'MIDI thru' option in the receiving application's 'Preferences' dialog or 'MIDI setup' dialog.

R.I.C.E. provides control over the MIDI channels in its MIDI output. Select **'Document settings...'** from the Edit menu and check the **Multi Channel MIDI output** option.

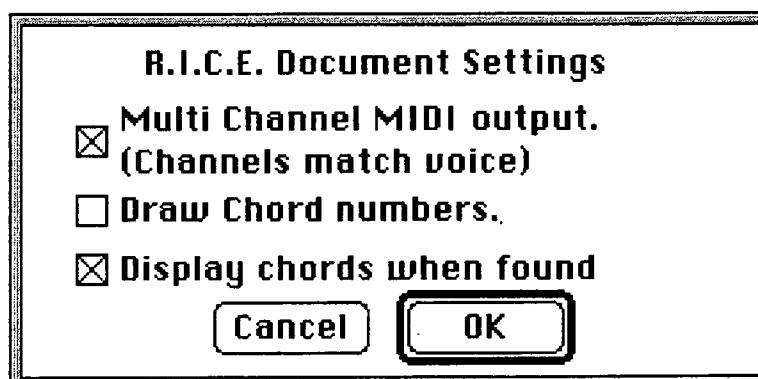
☒ **Multi Channel MIDI output.**
(Channels match voice)

This allows receiving programs that are capable of Multi-Channel recording to discriminate which notes belong to which voice by their MIDI channel. For example, this option is useful if you are routing your output to a program that will transcribe a quartet you are working on. The separate MIDI channels will allow that program to transcribe the parts individually.

Finale 3.x users need to note that Finale cannot record live MIDI data while running in the background, and hence it is not practically useful to connect R.I.C.E. to Finale via the IAC Bus. A satisfactory work-around is to perform your music into an OMS capable sequencer, quantise the performance and export it as a standard MIDI file for Finale to transcribe. This work-around also may be the solution to similar problems you may experience with other applications.

THE 'DOCUMENT SETTINGS...' DIALOG

A number of settings may be configured by selecting the **'Document Settings...'** under the Edit menu.



Multi Channel Output tells R.I.C.E. to match the MIDI channel with the voice number when playing chords. This is useful when receiving applications on the IAC Bus need to discriminate between R.I.C.E.'s different voices. See 'The 'Perform' mode' under 'Data output in this manual for more detail.

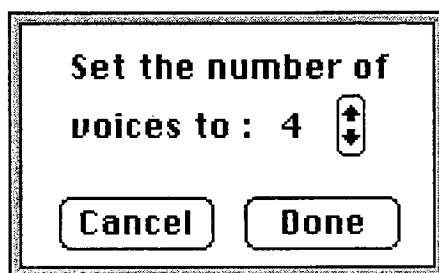
Draw Chord numbers forces R.I.C.E. to display the number of each chord in the Slide Strip. Normally this only happens when the Canon Manager Window is open. Note that screen redraws are significantly slower when this option is selected.

When searching for possible chords, R.I.C.E. usually displays and plays the chords as it finds them. Unchecking **Draw chords when found** suppresses this action and instead, R.I.C.E. displays the palette of possible chords *after* searching is complete. This option is available because displaying the chords as they are found during searching can significantly increase the overall search time on slower computers.

The '**Document Settings...**' are saved with every file.

SETTING THE NUMBER OF VOICES

R.I.C.E. has the capacity to work with one to eight voices. The number of voices can be set for each document. By selecting '**Set number of voices...**' from the Edit menu, you can change how many voices you work with.



You can alter the number of voices by clicking the 'up' or 'down' arrows. Note that the Display window is updated automatically after clicking the arrows. The interval restriction table windows resize themselves appropriately after the dialog is dismissed.

The 'Set number of voices' dialog is typically used after creating a new file. Changing the number of voices while working with an established file is quite possible, but not recommended unless you are sure of yourself. Increasing the number of voices can affect the contents of the interval restriction tables in a confusing manner.

When you decrease the number of voices, the settings for the voices that have been removed are hidden from the interval restrictions tables. They are not *cleared*, simply hidden. Increasing the number of voices to the original restores the original interval restriction tables, (assuming they were configured).

Increasing the number of voices fills the new voice's interval checkboxes with 'blanks'

Four Voices											
Vertical Intervals											
	0	1	2	3	4	5	6	7	8	9	10 11
1-2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3-4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Five Voices											
Vertical Intervals											
	0	1	2	3	4	5	6	7	8	9	10 11
1-2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1-5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2-4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2-5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3-4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3-5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4-5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

These 'blanks' *must* be filled in before the search engine will return anything. The blanks appear in all three of the restrictions tables, Linear, Vertical and Pitch. If the blanks are left blank, searches will always return nothing, because some of the voices have settings that permit nothing. This is what the blanks imply.

The decision to render new voices interval restrictions blank was made so that it would be easy for the user to spot a mistake in the tables, (which is what a blank amounts to). If the restrictions for each new voice had been filled, then the search engine you return potentially thousands of chords, (which is a problem in itself), but the user may not notice only a slight increase in possible chords, and be unaware that the tables were not configured as they should be. Imagine how annoying it would be to have worked for several hours with faulty restriction tables!

Because the search engine generates chords to check from the Linear Intervals table, the set of restrictions for newly added voices are not quite blank, and interval class zero is checked.

Linear Intervals											
	0	1	2	3	4	5	6	7	8	9	10 11
1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>


Not quite blank →

This is because the Search Engine requires at least one linear interval in each voice to function correctly. This is *not* an imposition on you, the composer, because without any linear intervals you voices would never move. If you *do* want *some* of your voices to remain static, select interval class zero, or alternatively, play into a voice a sequence of the same note, lock them, and harmonise them.

LIST OF SHIFT CLICKS AND ADDITIONAL FEATURES

The following is a list of 'Shift-Clicks' and other shortcuts to features that make working with R.I.C.E. much quicker and easier.

General additional features in the Display window :

- Holding down 'Shift, Option and Command, ()' while the mouse is over the display window draws a 'Octave box' which follows the mouse around. This box is always an Octave tall, and is very useful for writing music that is to be played on keyboards. By using this feature, you can quickly check if a passage of music will fit under a pianists hand.
- Shift-Clicking on a note, and then dragging up or down will transpose that note up or down an octave when the mouse is released. You will hear aural conformation of this alteration. Note that this does *not* work with notes under the Centre Control, nor with notes in the palette of possible chords.
- Clicking on a note, and then dragging to the left will play all the notes from that note in that voice to the edge of the screen. Click the mouse button again to stop the playback early.
- Shift-Dragging the Centre Control scrolls the display so the same chord remains under the Centre Control. Sometimes however, this is not possible. For example, if you are scrolled to the start of a document, and then Shift-Drag the Centre Control to the right, R.I.C.E. will tell you that it can't happen. Shift-Dragging past the 'End-of-file' marker also has the same result.
- Clicking in the Slide Strip moves the Centre Control to clicked position.
- Option-Clicking a note *permanently* deletes that note. This does not work for notes in possible chords, or notes under the Centre Control.
- Control-Clicking a note toggles it's locked or unlocked state. This does not work for notes in possible chords, or notes under the Centre Control.

Additional features in the Restrictions Tables windows :

- Clicking on the numbers at the top of the window play the appropriate intervals or pitch classes.
- When you Shift-Click on a checkbox in a restrictions table, and move (not drag) the mouse over other checkboxes while holding down the Shift key, R.I.C.E. forces all checkboxes under the mouse to the same state as the first one clicked. For example, if you uncheck a checkbox, all the checkboxes the

mouse moves over will also become unchecked. Shift-Option clicking a checkbox and then moving the mouse over other checkboxes *inverts* any checkboxes's current state.

Additional features in the 'Edit' mode :

- Pressing the cursor arrow keys on your computer keyboard will move any selected unlocked notes in the editable voices in the direction of the arrow. Pressing the up or down keys transposes the selected chords up or down by a semitone. Pressing the up or down arrow keys while holding down the Shift key transposes the selection by an octave. Using the left and right arrow keys moves the selection in the appropriate direction. Horizontal motion will erase unlocked notes, but will not overwrite unlocked notes.
- Typing voice numbers on the computer keypad changes the 'editable voices' without needing to use the popup menu. Pressing a key acts as a toggle, if the voice is not editable it becomes so, and appears on the popup menu display. If the voice is editable, pressing the voice's key number makes it non-editable, and the number is removed from the popup menu. This, of course, only works when R.I.C.E. is in Edit Mode.

Search engine features :

- Shift Clicking in the Centre Control performs a directional search.
- Pressing ⌘ during an exhaustive search will pause the search and give you the option to audition the chords found thus far.

THE SCROLL BARS, WINDOW MANAGEMENT AND NOTE SIZE

There are two scroll bars on the right hand side of the main display window. These control the height or 'thickness' of the notes. The top scroll bar controls the position of the upper 'virtual' boundary, and the bottom scroll bar controls the bottom 'virtual' boundary. These virtual boundaries are different from the actual boundaries of the display window, but are in the same position.

Clicking the inner two arrows of the scroll bars moves these virtual boundaries 'further apart', allowing more notes to be displayed vertically. However, since the actual boundaries of the display window remain fixed, the notes become thinner to accommodate the increased number.

Clicking on the outer arrows of the scroll bars has the opposite effect, the virtual boundaries are moved 'closer together' and there are fewer notes filling the the same space, and the notes get thicker. Experimenting with the action of these scroll bars is essential - and it quickly becomes intuitive.

This system with two scroll bars was employed so that you can always fit all the notes you want to see on the screen at any one time, regardless of the size of the display window. For example, scrolling the bottom scroll bar so see some very low notes will not affect any high notes which might otherwise have to be scrolled off the top of the window to fit in the low notes.

The width of the notes is controlled by the horizontal scroll bar at the bottom of the display window. This scrollbar normally scrolls the display. By Shift-Clicking on the arrows of this scroll bar the notes can be lengthened or shortened. (This change is purely visual). Shift-Clicking the right hand arrow shortens the notes (fitting more in the window), and Shift-Clicking the left hand arrow lengthens the notes. Of course, a normal click simply scrolls the notes.

The standard 'Zoom' and 'Grow' boxes in the right hand corners of the display window resize the window in a standard fashion. However, the behaviour of the *contents* of the window is non-standard. A typical graphics program or word processor leaves the data displayed at the same size and simply displays more or fewer notes according to whether the window was enlarged or shrunk. R.I.C.E., on the other hand reseizes the notes in the window proportionally to the change in size of the window. This is because in R.I.C.E., being able to make the notes larger quickly is more important than seeing more notes at once.

You can still resize the window horizontally and maintain the width of the notes. By Shift-Dragging the Grow icon (in the bottom right hand corner) you can force the notes to stay the same size. This allows more (or less) notes to be fitted horizontally on the

screen according to how you resized the window. Again, experimenting with these slightly unusual control is the best way of grasping how they work. After you have worked in R.I.C.E. for a while you will understand why they were implemented.

Note that when files are opened, the display window is set to the smallest possible size.

METHODS FOR WORKING WITH R.I.C.E.

UNDERSTANDING AND MANAGING THE SEARCH ENGINE

The Search Engine in R.I.C.E. does all the 'Hack Work'. It frequently examines hundreds of thousands of chords of which only a small percentage are returned to the possible chords palette. But how does it actually go about this?

As you know by now, the 'sound' of chords returned to the palette of possible chords is defined by the configuration of the interval restriction tables. In exhaustive searches, the Search Engine generates all possible chords from the chord under the Centre Control by moving each voice up and down in turn by every interval in the linear interval restrictions table. Every chord generated in this way is then checked against the harmonic restrictions table and the pitch class restrictions table. If the chord succeeds both these tests, it is included in the palette of possible chords. Directional searching only generates chords to be checked where the voices move in the directions you specify.

There are two critical factors which affect the overall time that an exhaustive search takes. Firstly, the length of a search is determined by the number of voices R.I.C.E. is working in. The number of harmonic intervals increases according to the formula $v_i = \frac{n(n-1)}{2}$, where v_i is the number of possible vertical intervals, and n is the number of voices. This increase in number is reflected in the size of the Vertical Interval restrictions table. Set the number of voices to eight to see how large the table gets.

The second factor which lengthens the time of exhaustive searches is the number of intervals switched 'on' in the Linear Intervals restrictions table. Increasing the number of linear intervals switched on increases the length of the search exponentially, according to a very complex formula. Increasing the number of vertical intervals that are switched on does not affect the length of a search anywhere nearly as dramatically.

Unfortunately, these two factors compound. Searching for possible chords using a large number of linear intervals above six voices takes a very long time, for example, well above half an hour on the fastest computers available.

There are a number of assumptions that the Search Engine makes when searching. The Search Engine only returns chords that have no repeating pitch classes. You will *never* get octaves or unison in a single chord in the palette of possible chords. This is

because octaves effectively reduce the independence of voices. Remember, this is a Restricted Interval *Counterpoint* Engine. The Search Engine will not return exact duplicate possible chords, but in the one pallet may (and often does) return possible chords that differ only in the octave placement of some of the voices. This is to facilitate composition of line and register.

The Search Engine currently ignores and consequently allows voice crossing. You must keep track of inter-voice interaction yourself.

The palette of possible chords is truncated when the number of chords found exceeds 1000.

MORE INFORMATION ON CONFIGURING THE INTERVAL RESTRICTION TABLES

Even though R.I.C.E. provides the 'Analysis Function' to assist with configuring the interval restriction tables, you may want to configure or extensively modify these tables yourself. Making changes or configurations that actually generate Possible Chords can be a baffling task.

Firstly, you need to note that often it is not enough to simply change the Vertical Intervals table if you want a change in harmony. Allowing an interval between certain voices is of no use if the linear intervals are not large or small enough to form that vertical interval! Appropriate changes typically must be made to both tables concurrently to achieve the desired result.

When setting up the Vertical Intervals table make sure that your selections make sense musically. For example, by selecting large intervals between adjacent voices and small intervals between outer voices, you are encouraging voice crossing, and this may be undesirable. Of course settings like this may work successfully, since R.I.C.E. does not distinguish between simple and compound intervals.

Secondly, setting up the tables according to patterns seems *not* to be a very successful method of configuration. Observe the results of the analysis function on some chords that you have played in and at first glance the results seem random or scattered, particularly in the Vertical Intervals table!

Make sure that after using the analysis function that all the tables contain what you really want. In particular check the Pitch Class restrictions table. The analysis function records in the Pitch Class table the pitches it finds in each voice. Typically however, you want to use more or all pitch classes in your newly generated material - not just the pitches that appeared in your analysis sample. Remember that in the Set Manager, you can change which restriction tables the analysis function acts on.

Be aware that the analysis function automatically fills the interval class zero checkboxes in the linear interval table, just to ensure that the search engine functions correctly if no other intervals were found. Generally this is of little practical value, and switching these checkboxes off can sometime significantly speed the Search Engine's progress, provided other intervals are checked 'on'.

RHYTHMIC STRATEGIES FOR R.I.C.E. ASSISTED COMPOSITIONS

You may have observed that R.I.C.E. holds no explicit rhythmic durational information. This means that R.I.C.E. does not differentiate between a note being a quaver or a semibreve or any other rhythmic denomination. Every chord and note in R.I.C.E. simply takes up one rhythmic unit. What every chord in R.I.C.E. represents is a *new harmony*, or alternatively a *single voices' movement*.

There are distinct advantages to working with music in this way, the main being that you can be freed from traditional notions of rhythm, meter and pitch until you want to consider them. You are working with changes in harmony only, the rhythm is abstract and pitch names are obscured. You are not bound to the restrictions that manuscript presents and implies, and you only have to deal with traditional notation at the latter stages of your compositional process, where it is appropriate to consider the mechanics of notation.

However, you will need to slightly change the way you work to get the most out of R.I.C.E. The main difference is that the order and importance of certain aspects of composition are altered.

The first (and initially most confusing) consideration is that if you are wanting to generate material that will eventually be polyphonic music, it helps to select possible chords that have notes *in common* with the chord you searched from.

This is because to generate music with independent sounding voices, voices need to move against each other, and this typically (but not always) means that one voice moves while another stays still. Selecting chords that have notes in common means that one voices is effectively 'staying still'. The other defining characteristic of independence of line is contrary motion.

Note that if you want to create music that contains block chords, you need to choose chords that have no notes in common.

The second rhythmic consideration is that repeated notes in a sequence may or may not be considered tied. At first this seems annoying to have all the notes with long durations repeated at each new verticality, yet tying notes together in a sequencer or notation package is generally easier than breaking them apart. When you drag across notes or use the 'Perform' mode, R.I.C.E. will play the entire chord as a new attack every time a voice moves, even though some of the notes may be virtually or conceptually 'tied'. This does sound strikingly different at first, particularly if you are hearing the chords through a percussive instrument, but one becomes quickly accustomed to it.

The third major consideration applies to harmonising preexisting material. A brief example here shows some possibilities. Imagine that you wanted to harmonise a Gregorian Chant, and you played the chant's notes into R.I.C.E. using the 'Insert Line' MIDI-in mode and locked them. Because each note takes up only one rhythmic unit, you would not be able to have harmonies moving under the longer notes in the chant. To circumvent this potential problem, simply repeat the note as many time as you plan to have harmony changes under it. Alternatively, use the 'Duplicate (or Remove) chord...' function if you change your mind as to the number of harmonies under a particular note during the harmonisation process. You are then free compose

moving harmonies over a static note being harmonised.

KNOWN BUGS

There are a number of known bugs in this beta release version.

- The 'Merge' operation in the Set Manager window does not function correctly.
- Large MIDI files are not entirely imported. The end is sometimes truncated.
- MIDI files with text markers cannot be imported. They cause R.I.C.E. to crash.
- Currently, only native R.I.C.E. files can be double clicked successfully. Double clicking other R.I.C.E. files, (or dragging TEXT and MIDI files onto R.I.C.E.) will cause R.I.C.E. to crash.
- The scrollbar action does not conform to the Macintosh interface standard.
- The possible chords scroll bar is not redrawn correctly on some occasions.
- R.I.C.E. will sometimes crash during performance mode. This bug is now lying low, and appears to come out only occasionally when the IAC bus is used.
- R.I.C.E. sometimes crashes when in the background, (usually behind a word processor!?) Typing Option-Command-esc normally successfully forces R.I.C.E. to quit without crashing other applications.

ANTICIPATED FEATURES AND IMPROVEMENTS FOR FUTURE VERSIONS

- Capacity to save the window positions and states.
- Faster search engine.
- Preconfigured libraries of interval tables defining a wide array of music.
- Proper Undo.
- Comprehensive MIDI channel configuration.
- A Morph function in the Set Manager Window to allow gradual transition from one set to another.
- A Contour Manager to guide R.I.C.E. chord selection when in 'Auto' Mode.
- A user configurable colour scheme for pitch and voice.
- Full feedback on the search engines progress, including the graphing of failures to pinpoint problem intervals in the restriction tables.
- An 'Instrument Manager' which checks the range and playability of each voice for specific instruments.
- An optional voice-crossing filter.
- Backfill and insert search modes.
- Comprehensive rhythmic support.

R.I.C.E. is very much a work in progress. The main intent of this Beta release is to gain feedback from potential users (composers!) about the functionality, potential and general 'usability' of this compositional tool. The ultimate aim of R.I.C.E. is to be a truly useful integrated composition tool, not just a 'curiosity piece'. *Your feedback on R.I.C.E. is essential.* The only obligation you are under after using this beta release of R.I.C.E. is to tell me what you think of it, how it could be useful to you, and how it could be improved. Where is it too restrictive? What can you imagine doing with it? I need to know!

I can be contacted at : jirrah@poboxes.com

The latest version of R.I.C.E., its documentation and R.I.C.E. compositions can be found at www.music.utas.edu.au/walker/software/rice/ricehome.html

Three Songs for Soprano and Piano

Text : Sylvia Plath

You're Morning Song The Night Dances

The next 18 pages have been removed for copyright reasons. They contain the 3 poems listed above and the music scores created for them. The music is copyrighted to Oxford University Press.

Music : Jirrah Walker

Summer 1998

Ten Prayers for Mixed Choir

From Michael Leunig's

Common Prayer Collection

Music : Jirrah Walker

Summer 1998

When the heart	SATB
God help us to live slowly	SATB
God bless the lost	SATB
God bless this tiny little boat	SATB
God help us if our world should grow dark	SATTB
Love one another	SSS
God be among us	SAATTB
God rest us	SATB
Let it go	SSATB
Dear God	SATB

The next 38 pages have been removed for copyright reasons. They contain the poems by Michael Leunig listed above, the music scores created for them, and drawings by Leunig.

Ten Prayers for Soprano and Strings

From Michael Leunig's
Common Prayer Collection

Music : Jirrah Walker

Summer 1998

When the heart

God help us to live slowly

God bless the lost

God bless this tiny little boat

God help us if our world should grow dark

Love one another

God be among us

God rest us

Let it go

Dear God

The next 22 pages have been removed for copyright reasons. They contain the music scores for the Michael Leunig poems listed above. The scores include the words of the poems.

Four Fugues

for

Two Pianos

I, II, III, IV

Music : Jirrah Walker
Autumn 1998

Arranged : Kim Bastin
December 1999

This music has been removed for copyright or proprietary reasons

Four Fugues

for

Mixed Ensemble

I, II, III, IV

Music : Jirrah Walker

Autumn 1998

Fugue I

[illegible]

15

Fl *f* *mp* *mp* *f*

Cl *mf* *mf* *f*

B Cl *mp* *f* *f*

S Sax *mp*

Bari *mp* *mf* *f*

Trp *mf* *mf* *f*

Tba *f* *f*

Pn *mp* *mf*

Vcl *mf* *no dim.* *mp* *mf* *f*

22

Fl

22

Cl

22

B Cl

22

S Sax

22

Bari

22

Trp

22

Tba

22

Pn

22

22

Vcl

The musical score for Fugue I, measures 22-25, is presented for a large ensemble. The instruments and their parts are as follows:

- Flute (Fl):** Measures 22-25. Measure 22 has a whole rest. Measure 23 enters with a half note G4 (f). Measure 24 continues with eighth notes. Measure 25 has a whole rest.
- Clarinet (Cl):** Measures 22-25. Measure 22 has a whole rest. Measure 23 enters with a half note G4 (f). Measure 24 continues with eighth notes. Measure 25 has a whole rest.
- Bass Clarinet (B Cl):** Measures 22-25. Measure 22 has a whole rest. Measure 23 enters with a half note G3 (f). Measure 24 continues with eighth notes. Measure 25 has a whole rest.
- Saxophone (S Sax):** Measures 22-25. Measure 22 enters with a half note G4 (f). Measure 23 continues with eighth notes. Measure 24 has a whole rest. Measure 25 continues with eighth notes.
- Baritone (Bari):** Measures 22-25. Measure 22 has a whole rest. Measure 23 enters with a half note G4 (f). Measure 24 continues with eighth notes. Measure 25 has a whole rest.
- Trumpet (Trp):** Measures 22-25. Measure 22 enters with a half note G4 (f). Measure 23 continues with eighth notes. Measure 24 has a whole rest. Measure 25 continues with eighth notes.
- Trombone (Tba):** Measures 22-25. Measure 22 has a whole rest. Measure 23 enters with a half note G3 (f). Measure 24 continues with eighth notes. Measure 25 has a whole rest.
- Piano (Pn):** Measures 22-25. Measure 22 enters with a half note G4 (f). Measure 23 continues with eighth notes. Measure 24 has a whole rest. Measure 25 continues with eighth notes.
- Violoncello (Vcl):** Measures 22-25. Measure 22 enters with a half note G4 (f). Measure 23 continues with eighth notes. Measure 24 has a whole rest. Measure 25 continues with eighth notes.

29

Fl *mp* *mp* *mp* *f* *mf*

Cl *mf* *mf*

B Cl *f* *f* *mp* *mp*

S Sax *mp* *mp* *mf*

Bari *mf* *f* *mf*

Trp *f* *f* *mf* *mf*

Tba *f* *mp*

Pn *mp*

Vcl *mf* *mp* *mf* *mf*

36

Fl *mf* *mp* *mf*

Cl *mp* *mf* *mf*

B Cl *mp* *f* *mf* *mp*

S Sax *mf* *mp* *mf*

Bari

Trp *mp* *mf*

Tba *mf* *mp* *mf*

Pn

Vcl *mp*

43

Fl

43

Cl

43

B Cl

mf

mp

mf

mf

43

S Sax

mf

43

Bari

43

Trp

43

Tba

mp

mf

mf

43

Pn

43

Vcl

Detailed description of the musical score: The score is for a piece titled 'Fugue I'. It consists of nine staves. The first staff is for Flute (Fl), the second for Clarinet (Cl), the third for Bass Clarinet (B Cl), the fourth for Saxophone (S Sax), the fifth for Baritone (Bari), the sixth for Trumpet (Trp), the seventh for Trombone (Tba), the eighth for Piano (Pn), and the ninth for Violoncello (Vcl). The score begins at measure 43. The Flute staff has a whole note G4 with a flat, followed by a whole rest. The Clarinet staff has a whole rest. The Bass Clarinet staff has a half note G3 with a flat, followed by a half note F3 with a flat, then a half note E3 with a flat, and finally a whole note D3 with a flat. The Saxophone staff has a whole rest. The Baritone staff has a whole rest. The Trumpet staff has a whole rest. The Trombone staff has a half note G3 with a flat, followed by a half note F3 with a flat, then a half note E3 with a flat, and finally a whole note D3 with a flat. The Piano staff has a whole rest. The Violoncello staff has a whole rest. Dynamic markings include *mf* (mezzo-forte) and *mp* (mezzo-piano). The score ends at measure 47.

Fugue II

Jirrah Walker
1998 $\text{♩} = 80$

Flute

Clarinet

Bass Clarinet

Soprano Sax

Baritone Sax

Trumpet

Tuba

Piano

Cello

1

mp *mp* *f* *mp* *mp* *f*

mf *f* *mp*

p *mf* *mf* *p* *mf* *mf*

mf

mf

7

Fl *mp* *mp* *f* *mp* *mp* *f*

Cl *mf* *f* *mf* *f*

B Cl *mf* *f* *mp* *mf* *mp*

S Sax *p* *mf* *mf* *p* *mf* *mf*

Bari *mp* *mf* *mp* *mf*

Trp *mp* *f* *mp* *mp* *f* *mp*

Tba *mf* *mf*

Pn

Vcl *f* *mp* *mf*

13

Fl *mp* *mp* *f* *f* *mp*

Cl *mf* *f* *f*

B Cl *mf* *f* *mp* *mp* *f*

S Sax *p* *mf* *mf* *mf* *mf*

Bari *mp* *mf* *mp* *mf* *mf* *mp* *mf* *mp*

Trp *mp* *f* *mp* *mp* *f*

Tba *mf* *mp* *mf*

8va

Pn *mp*

Vcl *f* *mp* *mf* *mf* *mp*

Fl 18 *mp* *mf* *f* *f* *mf*

Cl 18 *mf* *mp* *mf* *mp* *mf* *mp* *mf*

B Cl 18 *mf* *p*

S Sax 18 *p* *mf* *mp* *mf* *mf*

Bari 18 *mp* *mf*

Trp 18 *mf* *mp* *mp* *mf* *mp* *mf* *mf*

Tba 18 *mp* *f* *mf* *f*

(8va)----- 8va-----

Pn 18

Vcl 18 *mf* *f* *mf* *mp* *p* *mp*

23

Fl *mf* *mf* *f* *mf* *mf*

Cl *mf* *mp* *mp* *mp* *mf* *mf* *mp*

B Cl *mf* *p* *mf*

S Sax *mp* *mf* *mp* *mf* *mf* *mp* *mf*

Bari *mf* *mp* *mf*

Trp *mp* *mf* *mp*

Tba *mp* *mf* *f* *mp*

(8^{va})

Pn

Vcl *mf* *p* *mp* *mf*

27

Fl

27

Cl

27

B Cl

27

S Sax

27

Bari

27

Trp

27

Tba

27

Pn

27

Vcl

mf *f* *mf* *mf* *mf* *f* *mf*

mp

p *mf* *p*

mp *mf* *mf* *mp* *mf* *mp* *mf*

mp *mf* *mp*

mf *f* *mp* *mf* *f*

p *mp* *mf*

32

Fl

mf

mf

32

Cl

32

B Cl

mf

p

mf

p

32

S Sax

mp

mf

mp

mf

32

Bari

32

Trp

32

Tba

mp

mf

f

mp

32

Pn

32

Vcl

Fugue III

Jirrah Walker
1998

$\text{♩} = 80$

The score is written for a woodwind and string ensemble. The instruments and their parts are as follows:

- Flute:** Melodic line with dynamics *f*, *mp*, *mf*, and *mf*.
- Clarinet:** Rest.
- Bass Clarinet:** Rest.
- Soprano Sax:** Rest.
- Baritone Sax:** Rest.
- Trumpet:** Rest.
- Tuba:** Rest.
- Piano:** Rest (Grand Staff).
- Cello:** Melodic line with dynamics *f* and *mp*.

6

Fl *mp* *mf* *mf* *f*

Cl *mp* *mf* *mf* *mp*

B Cl *f*

S Sax *mp* *mf* *mp* *mf* *f*

Bari *mf* *mf* *mp* *mf*

Trp *f* *mf* *mf* *mf*

Tba *f* *mf*

Pn *f* *f* *mf* *mf*

Vcl *mf* *mf* *mp* *mf*

Detailed description: This is a page of a musical score for a piece titled 'Fugue III'. It contains measures 6 through 9. The instruments are arranged in a standard orchestral layout. The Flute (Fl) part starts in measure 6 with a melody, marked *mp*, then *mf*, and ends with a crescendo to *f*. The Clarinet (Cl) part enters in measure 6 with a melody, marked *mp*, then *mf*, and ends with a *mp* dynamic. The Bass Clarinet (B Cl) part is silent until measure 8, where it enters with a melody marked *f*. The Saxophone (S Sax) part has a melody in measure 6 marked *mp*, then *mf*, and a crescendo to *f* in measure 9. The Baritone (Bari) part has a melody in measure 6 marked *mf*, then *mf*, and a crescendo to *mf* in measure 9. The Trumpet (Trp) part has a melody in measure 6 marked *f*, then *mf*, and a crescendo to *mf* in measure 9. The Trombone (Tba) part is silent until measure 8, where it enters with a melody marked *f*, then *mf*. The Piano (Pn) part has a melody in measure 6 marked *f*, then *f*, and a crescendo to *mf* in measure 9. The Violoncello (Vcl) part has a melody in measure 6 marked *mf*, then *mf*, and a crescendo to *mf* in measure 9.

This page of a musical score is for a jazz ensemble, featuring ten staves for the following instruments: Flute (Fl), Clarinet (Cl), Bass Clarinet (B Cl), Saxophone (S Sax), Baritone (Bari), Trumpet (Trp), Trombone (Tba), Piano (Pn), and Violoncello (Vcl). The score is written in 4/4 time and includes various musical notations such as notes, rests, and dynamic markings like *mf*, *f*, and *mp*. The piano part (Pn) is written in grand staff notation, and the cello part (Vcl) is in bass clef. The score is divided into measures by vertical bar lines, and some measures contain multiple notes or rests. The dynamic markings are placed below the staves, indicating the volume or intensity of the sound. The overall style is that of a professional musical score, with clear notation and a structured layout.

15

Fl *f*

Cl *mf* *mp* *mf* *mp*

B Cl *mp* *mf* *mp* *mf*

S Sax *f*

Bari *mf*

Trp *mf* *mp* *mf* *mp*

Tba *mf* *mp*

Pn *mf* *mp* *mf*

Vcl *mf* *mp*

Detailed description: This is a page of a musical score for a piece titled 'Fugue III'. The page contains nine staves, each representing a different instrument. The measures shown are numbered 15, 16, 17, and 18. The instruments and their corresponding dynamics are: Flute (Fl) with *f*; Clarinet (Cl) with *mf*, *mp*, *mf*, and *mp*; Bass Clarinet (B Cl) with *mp*, *mf*, *mp*, and *mf*; Saxophone (S Sax) with *f*; Baritone (Bari) with *mf*; Trumpet (Trp) with *mf*, *mp*, *mf*, and *mp*; Trombone (Tba) with *mf* and *mp*; Piano (Pn) with *mf*, *mp*, and *mf*; and Violoncello (Vcl) with *mf* and *mp*. The notation includes various musical symbols such as notes, rests, and slurs, indicating the specific musical content for each instrument in each measure.

19

Fl

Cl

mp *mf*

B Cl

mp *mf* *f*

S Sax

Bari

mf *mp* *mf*

Trp

mp *mp* *mf*

Tba

mf *f*

Pn

f *mf*

Vcl

mf *mp* *mp*

23

Fl

23

Cl

f

23

B Cl

23

S Sax

23

Bari

mf

mf

23

Trp

f

23

Tba

23

Pn

23

23

Vcl

mp

mf

mf

f

Fugue IV

Transposed Score

Jirrah Walker
1998

$\text{♩} = 60$

Flute

Clarinet

Bass Clarinet

Soprano Sax

Tenor Sax

Trumpet

Tuba

Cello

mf

mf

mf

f

mp

6

Fl

mf

mp

6

Cl

mf

f

mp

6

B Cl

6

S Sax

mp

mf

mf

mp

6

T Sax

mf

6

Trp

mf

mp

6

Tba

6

Vcl

mf

11

Fl *mf* *mf* *mp* *mf*

Cl *mf* *mp* *mf*

B Cl *mf*

S Sax *mf* *mp* *f* *mf*

T Sax *f* *mp*

Trp *mf* *mp* *f* *mf*

Tba *mf* *f*

Vcl *mf* *mp*

This page of musical notation, numbered 11, is arranged for a jazz ensemble. It contains seven staves for woodwinds and brass instruments, and one staff for the Violoncello (Vcl). The instruments and their parts are as follows:

- Flute (Fl):** The first staff, starting with a treble clef and a key signature of one sharp (F#). It features a melodic line with dynamic markings of *mf*, *mf*, *mp*, and *mf*.
- Clarinet (Cl):** The second staff, also with a treble clef and one sharp. It has a melodic line with dynamic markings of *mf*, *mp*, and *mf*.
- Bass Clarinet (B Cl):** The third staff, with a treble clef and one sharp. It contains a few notes with a *mf* dynamic marking.
- Soprano Saxophone (S Sax):** The fourth staff, with a treble clef and one sharp. It features a melodic line with dynamic markings of *mf*, *mp*, *f*, and *mf*.
- Tenor Saxophone (T Sax):** The fifth staff, with a treble clef and one sharp. It has a melodic line with dynamic markings of *f* and *mp*.
- Trumpet (Trp):** The sixth staff, with a treble clef and one sharp. It features a melodic line with dynamic markings of *mf*, *mp*, *f*, and *mf*.
- Tuba/Euphonium (Tba):** The seventh staff, with a bass clef and one sharp. It has a melodic line with dynamic markings of *mf* and *f*.
- Violoncello (Vcl):** The eighth staff, with a bass clef and one sharp. It features a melodic line with dynamic markings of *mf* and *mp*.

The notation includes various musical symbols such as notes, rests, and dynamic markings (*mf*, *mp*, *f*). The page is numbered 11 in the top left corner.

16

Fl *mp*

Cl *mp*

B Cl *mf*

S Sax *mf* *mf*

T Sax *mf*

Trp *mf* *f*

Tba *mf*

Vcl *mf*

26 *mf* *ff* *mf* *f* *ff*

26 *mf*

26 *f*

26 *mf* *mp* *mp*

26 *mf* *f* *mf*

26 *mp* *mp*

26 *f*

26 *mf* *f*

Fl

Cl

B Cl

S Sax

T Sax

Trp

Tba

Vcl

This musical score page contains eight staves, each for a different instrument. The Flute (Fl) staff starts with a melodic line marked *mf*, followed by a *ff* section, then *mf*, *f*, and *ff*. The Clarinet (Cl) staff has a rest followed by a *mf* entry. The Bass Clarinet (B Cl) staff features a *f* dynamic. The Soprano Saxophone (S Sax) and Tenor Saxophone (T Sax) staves have *mf* and *mp* markings. The Trumpet (Trp) staff has *mp* markings. The Trombone (Tba) staff starts with a *f* dynamic. The Violoncello (Vcl) staff has *mf* and *f* markings. The score includes various musical notations such as eighth notes, quarter notes, and rests, with dynamic markings (*mf*, *ff*, *f*, *mp*) and articulation marks.

31

Fl *mf* *f* *f*

Cl *mf* *mp* *mp* *mf*

B Cl *f* *mf*

S Sax *mf* *mf* *mf*

T Sax *mf*

Trp *mf* *mp* *mf* *f*

Tba *mf* *mf*

Vcl *mf* *mf* *f*

Detailed description: This is a page of a musical score for a jazz ensemble, covering measures 31 through 34. The instruments listed on the left are Flute (Fl), Clarinet (Cl), Bass Clarinet (B Cl), Soprano Saxophone (S Sax), Tenor Saxophone (T Sax), Trumpet (Trp), Trombone (Tba), and Violoncello (Vcl). The key signature has one sharp (F#), and the time signature is 4/4. Measure 31 begins with a rehearsal mark. The Flute part features a melodic line with dynamics *mf*, *f*, and *f*. The Clarinet part has a similar melodic line with dynamics *mf*, *mp*, *mp*, and *mf*. The Bass Clarinet part enters in measure 32 with a *f* dynamic. The Soprano Saxophone part has a melodic line with *mf* dynamics. The Tenor Saxophone part has a melodic line with *mf* dynamics. The Trumpet part has a melodic line with dynamics *mf*, *mp*, *mf*, and *f*. The Trombone part has a melodic line with *mf* dynamics. The Violoncello part has a melodic line with dynamics *mf*, *mf*, and *f*. The score includes various musical notations such as eighth notes, quarter notes, and half notes, as well as slurs and dynamic markings.

36

Fl

mp *mf* *mf* *f*

Cl

f *mf* *f*

B Cl

mp *mf* *f*

S Sax

f *mp* *mf* *mf* *f*

T Sax

f *f* *f* *f*

Trp

mf *f*

Tba

f *mf* *f*

Vcl

f *f* *f* *f*

Detailed description: This is a page of a musical score for a jazz ensemble, covering measures 36 through 40. The score is written for eight instruments: Flute (Fl), Clarinet (Cl), Bass Clarinet (B Cl), Soprano Saxophone (S Sax), Tenor Saxophone (T Sax), Trumpet (Trp), Trombone (Tba), and Violoncello (Vcl). The key signature has one sharp (F#), and the time signature is 4/4. The music is characterized by complex, often chromatic, melodic lines. Dynamics are indicated by letters: *f* (forte), *mp* (mezzo-piano), *mf* (mezzo-forte), and *f* (forte). Slurs and hairpins are used to indicate phrasing and volume changes. The Flute part starts with a rest in measure 36, then enters in measure 37 with a *mp* dynamic, rising to *mf* and *f* by measure 40. The Clarinet and Bass Clarinet parts have more active lines, with the Clarinet starting at *f* in measure 36. The Saxophone and Trumpet parts also show dynamic shifts, with the Soprano Saxophone starting at *f* and the Trumpet at *mf*. The Tenor Saxophone and Trombone parts are more sparse, with the Tenor Saxophone playing *f* notes and the Trombone entering in measure 39. The Violoncello part provides a rhythmic and harmonic foundation, starting at *f* and maintaining it throughout.

Three Reharmonised Jazz Standards

for

Jazz Quartet

All the things you are
Here's that Rainy Day
Watermelon Man

Hammerstein and Kern
Van Heusen and Burke
Herbie Hancock

Harmonisations : Jirrah Walker

These 3 scores have been removed for copyright or proprietary reasons.

Summer 1998